

THE PAPER

FOR OWNERS OF THE COMMODORE PET™ PERSONAL COMPUTER

VOLUME III, ISSUE 6/7

A CSD-LIPS PUBLICATION

\$2.50

In This Issue

General Info		2
Foreground Background	pub/ed	3
DASM Review	Ralph Bressler	5
Reader I/O	pub/ed/readers	6
Not So Random RND	Alex Breed	9
Puzzle Solutions	Bill Batcher	10
Micros in Psychology	George Skilich	11
Observations: Issues 1-3	Roy Busdiecker	12
Meddling with Middle Strings	Gerry Eisner	14
SEARCH Note	James MacArthur	15
A Better Screen Copy	J. R. Bromley	16
Cassette Survival Hints	D. Sherwood	18
ASSM/TED Review	J. Fowler	20
Now You See It	Ralph Bressler	21
Auto Repeat Keys: Version 2	Roy Busdiecker	22
Program Exchange	C. Deschamps	24
Program Exchange List		25
Observations Resumed	Roy Busdiecker	30
Auto Data Entry	Ralph Bressler	33
LIPS Contest Winners		35
Screen Dump	B. V. Haluschak	36
Assembly Language: Part II	J. Fowler	37
SAT/PSAT Review	S. Silverman	39
ML Relocator	A. Wachtel	40
Speed-up Explanation	W. Swan	40
Comments on Issue 4/5	Fran Turco	41
Response to Fran Turco	pub/ed	42

User's Groups

We recognize that local user's groups are the most important aid to both the beginning and experienced user. We will publish the name, contact person and address for any user's group. Please send us this information for any group you know of. We can also print meeting times or special announcements or requests. Please remember to send updates as this information changes.

See back cover for LIPS contest

General Information

The PAPER is published 10 times per year by Centerbrook Software Designs and the Long Island PET Society at 98 Emily Dr., Centereach, NY 11720. Telephone: (516) 585-2402.

The PAPER will be mailed to subscribers near the end of the month until the volume is completed. New subscribers will receive all issues of Volume 3. Single copy price is \$2 and the subscription price is \$15 for all the issues of the current volume. Subscription orders should be mailed to The PAPER, Box 524, East Setauket, NY 11733.

Third class postage is paid at East Setauket, NY 11733 (Permit #96). POSTMASTER: Mail all address changes to the address above.

The PAPER, Centerbrook Software Designs and the Long Island PET Society are in no way associated with Commodore Business machines. CBM is not responsible for any of the contents of The PAPER unless otherwise noted. PET and CBM are trademarks of Commodore Business Machines.

All readers are encouraged to submit articles of general interest to PET users. Materials submitted must be free of copyright restrictions. All contents of The PAPER are not copyrighted. All articles remain the property of the author and may be reprinted with their permission. Please include a note stating that the article was originally published in The PAPER.

Subscription Rates:

USA third class: \$15/volume

Non-USA first class surface: \$20/volume

Payment in check or money order in US funds must accompany all orders. Only prepaid purchase orders will be accepted. All checks should be made out to The PAPER. Sorry, we cannot accept bank or credit cards.

Advertising

Advertising rates are \$25 per quarter page per issue. Copy must be camera ready or there will be an additional charge. Special rates may be negotiated.

Circulation

Volume 3 Issue 4/5:

Printed: 1000 USA: 526 Canada: 44 Foreign: 41

Software Exchange

Software published in The PAPER or distributed through the Exchange is believed to be free of copyright restrictions. It is meant to work on the machine indicated. Many programs were originally designed for the old ROM PETs but efforts have been made to convert them to work on BASIC 1.0 and 2.0. Write for a complete list of programs.

Staff:

Publisher: Ralph Bressler

Editor: Doug Haluza

Assoc. Ed: Roy Busdieker

Vic SantaLucia

Exchange: Charlotte Deschamps

Staff writers: Bill Batcher

JoAnn Comito

Gerry Eisner

Jim Folwer

Foreground Background

by Raleh and Doug

After one issue many things have become clearer than they were before. Its hard to do everything by yourself. We type, edit, proofread and paste up the whole issue. I suspect this is more due to the fact that we are disorganized than the lack of volunteers. We have to thank all those who wrote letters or articles since it would be impossible for us to write all the material too. Also, thank you to the readers that accepted my invitation to join us for Volume 3. We took over Volume 3 in the middle with approximately a \$4,000 commitment and a balance of \$150. Right now we need our subscribers to encourage others to subscribe. It would also be helpful if you convinced your favorite dealer or PET product maker to give us an ad.

POLICY

We will try to be consistent about our newsletter policies. Here are some things we believe.

Back Issues

ARESCO told me that Volumes I and II are available from them in limited quantities. When a person subscribes to the current volume, Volume 3, they get all the issues of that volume.

Programs

ARESCO will maintain some of the programs on the Software Shelf. We have none of these programs. Programs on exchange ARE available and some of these are really good. See the article in this issue for more information.

Articles

We welcome all articles from all readers and will publish any article. We reserve the right to make small editorial changes but the article remains your property. If you can submit articles in WordPro II or WordPro III format this would be helpful. Readers have said they would like to see articles on using the disk for data files and the magazines which deal with PET programming.

Letters

The only letters we will not publish are those that simply compliment us or those that severely criticize others. We do not intend to get involved in personal arguments.

Reviews

So many products are available for the PET today that a user has trouble making decisions. Many of these products have prices in the \$50 to \$150 range. We ask that people review only products they have experience with and that they not review products they have a special interest in.

(continued next page)

MITA

The microcomputer industry has its own fledgling trade organization, Microcomputer Industry Trade Association. The problem here is that the membership fee of \$200 may represent .01% of some company's income while it is 10% of another's. John Crais mentioned graduated fees at the August 1980 PCC. This is definitely what is needed if the small product suppliers are to have their voice.

Computer Shows

Recent computer shows have been disappointing. The Personal Computer Conference in August in Philadelphia had many small suppliers but few of the major manufacturers. Commodore, who recently moved to Norristown, failed to show. The National Small Computer Show in October at the New York Coliseum had the big guns but few small vendors. The prices were too high! A controversy has arisen as to the place of microcomputers at major, national computer shows. Big microcomputer companies like Apple and Commodore want to be in the big shows next to IBM and Hewlett-Packard. Small vendors will be closed out by the higher prices of the larger shows. Should micros be part of the large show or be a separate display? How many regional and national shows can there be without diluting the enthusiasm of the public and the economic resources of the vendors? What should the emphasis be in these shows; personal, educational or business?

Chips and Copyrights

The unending copying of some software has led to extreme protection methods by those companies that produce expensive, serious software. Some companies have begun protecting their disks so that they may not be copied even for backup. These companies charge a large percentage of the purchase price for replacement copies. Many new packages require a chip which usually contains no code but is just for protection. This increases the price of the package and uses up most of the empty ROM slots. Even Spacemakers or Dial-A-ROMs may not help. WordPro, VisiCalc and BPI are just a few interesting packages that require a chip to operate. Since it is impossible to keep all people from copying these programs and since it only takes a few copies to hurt business, it is hard to see a solution to this problem. Companies who protect their software should recognize that they have certain responsibilities. People who use serious, high-priced software are serious. They are using the programs for business applications and down time from ruined disks costs money. Companies must provide means to make backups or extremely fast and inexpensive replacement service. Software vendors who feel they must use chips for protection should include useful code in them. Dummy chips will soon be defeated by ardent copyright violators. ROMs with useful code will afford the company more protection and the user more memory space to work with.

Users Group

If you don't have a PET or micro users group in your area, I encourage you to start one NOW. User's groups can help those just getting started and provide an exchange of information on a personal level impossible in major magazines or even The PAPER. The Long Island PET Society started as about 20 teachers interested in using the PET. Two years later we number 150 and have a national newsletter.

DASM/TASM Package

A Product Review

by Ralph Bressler

Type: Software

Model PET: any model (DASM requires disk and new ROMs)

Source: Baker Enterprises
15 Windsor Drive
Atco, NJ 08004

Price: \$15 - disk, \$10 - tape

This is the first in a series of reviews of software written by Bob Baker. Bob writes the Petpourri column for Microcomputing and his company sells software written by Bob. He also distributes some PET supplies such as disks and printer ribbons at attractive prices. There are several reasons why I plan to review Bob's programs and why I would like to review them ALL NOW. First, Bob's programs are carefully and professionally done and they work! Second, many are unique in what they do and make use of interesting properties of the PET. Lastly, Bob's programs are the cheapest around. The package reviewed here is about the most expensive!

The Disk Assembler and Tape Assembler packages contain a complete editor/assembler/loader package for the PET using modified mnemonics and special source code formatting. These changes from standard assembly coding allow the assembler to assemble faster even though it is written in BASIC. The disk based version includes copies of BOTH tape and disk based programs plus several utilities. The tape based version contains only the tape programs, a full twelve pages of documentation comes with the package, as well as several demonstration programs.

The Editor program provided is easy to use and oriented toward assembly programs. It uses numbers to refer to lines of code and makes skipping from field to field simple. It can create, edit and print source files to the printer. The Editor helps to check format before running the assembly program through the Assembler. This helps to cut assembly time to 1 to 2 seconds per line of code.

The Assembler is a two pass assembler written in BASIC. On the first pass the symbol table is built and the value of each label is computed. On the second pass an object file is written to tape or disk and an assembly listing may be printed. The object file must then be loaded by the Loader program and then executed as you wish.

The listings provided are clear and include the memory location and object in HEX. This is followed by the source file line number and the line of source code. The source code includes a label, command, operand, and comment. Full line comments are also permissible. Listings may also be directed to the printer for later listing. The symbol table is also output at the end of the listings.

As far as any disadvantages so, I really couldn't find any serious ones for beginning programmers. As stated some special modifications have been made to the mnemonics and format. Also, the assembler does not support macros. This fact would not upset the beginner at all.

In summary, I may not be the best person to review this program since my experience with assembly language is very limited. However, as a beginner this package has really been a big help. I feel anyone with a little knowledge of assembly and a will to learn can use this package. Also, you won't find a faster, workable assembler anywhere.

Reader I/O

Reader I/O is your chance to ask questions and make short comments. Letters will be edited to get at their essence and then printed. We will answer what we can if an answer is required. Comments requiring no answer will also be printed. Questions we cannot answer will be included in the hope that readers can help each other. If you can answer a question or provide more information than we have, please write us and we will include your information. Many letters we recently received offered thanks, congratulations and wishes of good luck. We hope we can satisfy the needs of as many people as possible and live up to your expectations.

Bruce Karsol writes about a problem with reading files using BASIC 2.0. If one of the records has to be enclosed in quotes to allow a comma to be included, PET would respond with a ?FILE NOT OPEN ERROR when trying to read back that record. For some reason location 593 (# of files open) was being set to 0 after reading this string. His solution was to POKE 593,1 before using INPUT#1,T\$.

I ran into a similar problem when adapting a commercial word processor to work with disk. My solution was to limit string length to less than 80 characters which also seemed to work. You might also use GET#1 instead of INPUT#1, but this takes more time.

R.I. Laver has trouble with Doug Hennig's "Speeding Up the Print Statement". With the new ROMs both Z and Z1 in line 80 give values of 0. Even the corrections offered do not help. In fact, a direct POKE 59458,62 leaves some flickering spots on the screen and locks out the keyboard. At this point the only way to recover is to reset by turning the PET off.

You are not alone. This command has a similar effect on many PETs and this is why it is not recommended. See Warren Swan's article in this issue.

Jim Yost, Box 556, Sommerville, MA 02143, contacted me some time ago offering his help. I apologize for not responding sooner and in person. Jim has both old and new ROMs and considerable information on both. Jim sent copies of some screen dumps from his BASE2 printer using his own program. The resulting pictures are quite impressive and I'm sure he can provide valuable information on that device. He also writes that we should try the following line on our new ROM PETs.

```
FOR A=65457 TO 65468: ?CHR$(PEEK(A)): NEXT
```

Jim suggests that what appears was done just to preserve Commodore's 'not-quite-perfect' image.

We need all the help we can get. Right now articles are what we need the most. A review of the BASE2 or other things you have done would be interesting.

Gary Stone wants us to know about the Parallel Interfacing Element (PIE) made by LEMDATA, Box 1080, Columbia, MD 21044 and priced at \$89.95. This printer interface plugs directly into the PET's parallel user's port. A printer like the PR-40 can then be plugged into the PIE and since the PIE extends the edge connector of the parallel port this is still available to the user. The PIE provides a DIP switch selectable IEEE device number and has a PET style IEEE edge connector.

This interface also has an optional ROM which corrects the upper/lower case problems for the various PETs.

Thanks for the info. I'm sure many users would like to use other printers beside those from Commodore. This provides a possible solution for them.

Jay Chin needs help with interfacing a CDC CR134-D printer to the PET IEEE bus. He has some manuals but no schematics. Jay has already interfaced a PR-40 printer, an IBM Selectric typewriter and a Centronics 306 printer. He will share what he learned with anyone who contacts him at 1169 14th St., Upland, CA 91786.

I know your request was made some time ago but here it is. Good luck and thanks for the offer. I'd like to see an article relating some of your experiences and your evaluation of the various printers.

John Schueler has some suggestions which may extend the life of our PETs. He operates his PET and peripherals at 2% below rated line voltage using a Variac and line voltage meter. He also has a GE line voltage spike suppresor in the line buss which feeds the equipment. John's system is an old ROM 8K PET with an added 24K of memory, a Computhink disk, a PAICs Toolkit, a CBM 2023 printer and a Skyles big keyboard. He is pleased with the disk which loads and saves 10K programs in 1 sec.

I've had a PET an peripherals for 2 years and have had two brown-out problems and no other apparent problem. Your precautions seem like overkill but I'm sure you know more about it than I do. Many people seem to think that only the Commodore peripherals work with the PET. Your comments on the Computhink drive would be appreciated.

Emil Volcheck says that the Screen Print on page 39 of March/April '80 works well if it is placed in the second cassette buffer. He states that problems arise when it is placed in the top of memory since BASIC will write over it. His solution on the 32K PET is to POKE 52,134: POKE 53,127

This problem occurs with any routine stored in the top of memory. The top of memory pointer should always be set down just the amount needed to accomodate the routine. Be sure to remember to reset it.

Scott Summer wants to know if Commodore makes a 2040A single disk drive. He says that most dealers say no but that Advanced Computer Products has one advertised. ACP even said they had some more coming, quoted a price and gave specs. He also questions the availability of the Commodore electrostatic printer.

This business is full of announced but never produced products. As far as I know, Commodore never produced the single stripped drive. ACP may have a single drive, but I don't think its Commodore's. It should be stated that there are other fine drives compatible with the PET. Again, Commodore never produced their advertised electrostatic printer. Several companies beat them to it with good printers and I guess they didn't think it was worth it to produce their own. Two companies that did produce electrostatics are Skyles and Axiom.

Harry Calhoun needs information on the General Instruments AY3-8910 sound generator. He has been unable to get information or data from local dealers or from the company itself. The BYTE article from July 1979 is not sufficient. You can contact Harry directly at RT 1 Box 55, Medical Lake, WA 99022. Harry also says that dealers in his area do not support the PET but push competitive systems.

Let's face it, some people are just plain ignorant. This seems to include the dealers in your area. Have you tried to educate them? Advanced Computer Products is selling a manual for the GI chip for \$3.00. We have seen the manual and it is complete and should solve your problems. The 60 page manual and chip are advertised on page 10 of the 1980 catalog. ACP, PO Box 17329, Irvine, CA 92713.

Dan Condon asks us to 'get a better title' since many publications use PET without any problems. He wants us to let people know what our publication is about. Dan implores us to avoid 'shop talk' and in-jokes which do little to inform. He also would like to see more machine language programs which perform simple and interesting tasks and then explain why.

I think you provide some challenges to many of us. First, I really haven't thought much about the name. Let's have some ideas on this! We will try to avoid 'inside jokes and comments' and will try to explain what we are doing. At least you'll know that any unexplained comments or ideas are not on purpose but simply oversights. Your last comment is a challenge to our readers who understand and write machine language. The challenge, as I see it, is to provide short simple routines which are explained in such a way that all can understand.

R.L. Leutzinger needs information on how to connect his Heath H14 printer to his PET through the CMC ADA1400. When he tries this the buffer overloads and printing stops.

This is why some people, like me, bought only Commodore peripherals. It's not because Commodore products are the best but because we can be relatively sure they will work when plugged in. Therefore, I applaud your efforts and hope some reader can help. Have you tried contacting Connecticut microComputer directly?

John Bosak is suffering from lack of basic information about the PET. He comments that many articles in The PAPER have been of little value for his application or were beyond his level of understanding. John also wonders what Commodore should have supplied to PET owners and if he missed something.

You are not alone in your bewilderment at the lack of information provided by Commodore. Fortunately, many fine books and magazines now have PET related information. To discuss all of these now is beyond the scope of this column. I hope The PAPER will begin to provide information which you find useful. Let us know what you would like to see. Back to what Commodore owes us! When I originally got a PET during the summer of 1978, I got a small pamphlet explaining how to turn on the PET and make a rocket blast off. That was it and I never have gotten anything else for buying that PET. Later Commodore began supplying a 'manual' for the PET which contained basic information buried among references to ML programming and operating systems. This book is so unorganized that it is next to useless. Surprisingly, the manuals provided with peripherals (disk, printer) are informative and well-written as is the manual provided with the business machines. The later, however, contains little information for a real programmer. With the very newest machines Commodore is supplying a BASIC 4.0 manual, the business system manual and the new PET/CBM Personal Computer Guide published by Osborne. This is a great book. I think anyone who did not receive proper documentation for their PET (all of us) should write or call Commodore and request this new documentation.

That's the end for now. Please write. We promise to answer.

PET's Not-So-Random RND

by Alex Breed

One sees occasional references to the PET's random function producing a repeating series of random numbers. On my new ROM machine the random loop is delayed entry. From a cold start my computer generates 59022 random numbers that do not repeat, and then it begins a series of 13379 random numbers which does repeat indefinitely.

I have never tested an old ROM PET but would guess that its random number function would be similarly repetitive, perhaps with a different loop length and entry point. A PET will probably behave as mine does if these are the first two random numbers it produces from a cold start: 974626516 and 922007304.

On PETs where this is the case, a short program can be run which will verify the loop length and entry point. First, turn off the machine for about ten minutes to ensure a cold start. For some reason, just flicking the power off and on does not reseed the random number generator to its cold start value. Now run the program which takes about ten minutes. At the end of the run the screen will display the random numbers 59014-59033 and 72393-72412, and the point where the repetition begins will be evident.

```
10 PRINT "(clr)";
20 FOR N=1 TO 59013:X=RND(1):NEXT
30 FOR N=59014 TO 59033:PRINT N;RND(1):NEXT
40 FOR N=59034 TO 72392:X=RND(1):NEXT
50 PRINT "(home)";
60 FOR N=72393 TO72412:PRINTTAB(18)N;RND(1):NEXT
```

This repetitiveness obviously affects only programs that generate a great number of random numbers. I discovered the problem in a betting game which produced about 10,000 random numbers in each run. After a while I was getting nearly the same results in every game.

It does no good to attempt to avoid the loop by seeding the random function with $X=RND(0)$ or $X=RND(-TI)$. On the contrary, tests I have made indicated the loops are often shorter and are entered earlier than without seeding. Here is a BASIC subroutine which is slower than the machine function but avoids the loop problem. Since I am not a statistician, I have not rigorously tested it for distribution. It is based on the simple idea of multiplying two numbers together and using the fractional part of the product as the random number. The basic idea had to be altered a bit because in its simple form the routine also produces a loop. Loops are not easy to avoid in random generators as the people at Commodore should know.

First, in order to seed the subroutine, place this instruction somewhere in the main program where it will be executed only once and before the random subroutine is called:

```
10 R1=RND(-TI): R1=RND(1): R2=10000*RND(1)
```

Then there is the random subroutine:

```
50000 IF F1 THEN R2=R2-R1:F1=0:GOTO 50020
50010 R2=R2+R1:F1=1
50020 R1=R1*R2-INT(R1*R2):RETURN
```

This generates a random number, R1, which is a decimal fraction.

Puzzle Solutions

by Bill Batcher

In the last issue, I described how a puzzle developed in my class and how, unexpectedly, my students arrived at three different solutions. Not wishing to deprive you of the joy of discovery, I did not give the solutions at that time. Since that article was written, a fourth solution has been found. If you missed the last issue or forgot the puzzle, the problem was this:

- 1 - Type the following program into a PET:

```
10 INPUT A$
20 PRINT RIGHT$(A$,4)LEFT$(A$,LEN(A$)-4)
30 GOTO10
```

- 2 - Run the program and when the ? prompts you, type in a set of letters such that when you press RETURN the PET will respond with JIMMY CARTER or GEORGE WASHINGTON (or whatever).
- 3 - Once you have the idea, then comes the real puzzle. Try to get the computer to respond with ABE LINCOLN. The A must appear directly under the ? and there must be a space between ABE and LINCOLN.

If you haven't tried the puzzle, you should do so before looking at the solutions. Line 20 tells the PET to move the last four characters of the input to the beginning. So, of course, if we type LINCOLNABE<sp>, we have solved the problem. Unfortunately, when a space ends an input the PET drops it. We can type LINCOLNABE with 20 spaces after it and the PET will still interpret it as only 10 characters long. Three groups of students found ways to force the PET to keep a final space after the E.

Solution 1: Enclose your input in quotes: ? "LINCOLNABE "

The PET will interpret all 11 characters in the quotes as part of the input. Knowing this might also come in handy in situations where you wanted to include a colon or comma in your input. Without quotes this would give an 'EXTRA IGNORED' message. For instance, if three children wanted to play HANGMAN together they could type "AL,ED,BOB" when asked for a name.

Solution 2: Use reverse field after the E: ? LINCOLNABE(rvs sp)

The PET does not interpret 'reverse field space' the same as a 'space'. In fact, while space has a poke value of 32, reverse field space has a value of 160. So again, the PET would count 11 characters in this input. Whenever the PET goes to a new BASIC statement, it eliminates the reverse field, so the space after the E would be printed normally. In fact, another solution would have been to type the entire line in reverse field.

Solution 3: Use shift space after the E: ? LINCOLNABE(shift space)

Again 'shift space', poke value 96, is not the same as space. It even has a different ASCII value. So the PET will again count all 11

characters. This 'shift space' also affords an extremely simple routine for eliminating the sudden endings of programs. Here is a program line which will prevent the ending of a program if RETURN is prematurely hit:

```
100 INPUT"TRUE OR FALSE<3 shift space><3 left cursor> ";A$
```

Finally, a teacher approached the problem differently. Instead of asking 'How can I hold the space after the E?', she simply place a space before the L and held it with a left hand quote.

```
? " LINCOLN ABE
```

But then the A was'nt under the ?, so she asked 'What character can be put before th A which will not consume a spot on the screen ?'.

Solution 4: Use an INSERT before the A: ? " LINCOLN<insert>ABE

Actually, this solution requires a 12 character input but one character is a NULL character. The 'insert' does nothing except set itself counted. Another null character that could have been used is the OFF key. Null characters might be handy if you wanted to treat strings of different lengths as if they were all the same length.

This concludes the solutions to this puzzle. It is clear that the investigation of this puzzle led to the discovery of many different, interesting and useful ideas. Please write if you find other solutions. Since these kinds of problems seem to stimulate interest and lead to unexpected discoveries, it might be nice to make puzzles a regular feature. Let us know if any other puzzles develop.

Microcomputers in Psychology

by Dr. George Spilich

For the last twenty years, psychologists have increasingly used computers in their work. This utilization has taken a number of forms; for example, computers have proven useful in data reduction, in simulation of cognitive processes, etc. Microcomputers have been found to be extremely useful in laboratory situations in psychology; typically they are used to provide visual stimuli to subjects and to record their responses. The latency of such responses can also be measured. These small computers can also be used to teach undergraduates about the nature of cognitive operations.

I suspect that there are a number of psychology labs around the country using micros such as the PET or Apple as a research or didactic tool, or both. At Washington College, we are at present using an 8K PET as a demonstrator of basic paradigms in cognitive psychology and also as a programmable device for students to use in their own research projects. We are about to upgrade our PET with additional memory or a disk and other peripherals. I would like to hear from anyone also using micros for similar purposes in cognitive psychology in order to set up a specialized PET user's group. Through this group we could all share in answers to common problems. This group could be interdisciplinary in the field psychology. Anyone interested should contact me c/o Psychology Department, Washington College, Chestertown, MD 21620. Please indicate your interests, needs and types of computers.

by Roy Busdiecker

(Pub Note - I must apologize to Roy. He sent his 'Observations' in right on time. In the transition from ARESCO to us their publication has been delayed. I thought many of Roy's comments so valuable that I have included them here in a condensed form.)

Most significant, in my opinion, is that Volume 3 is being published. With over two years experience with PETs behind us, there is an awesome amount of information behind us. This is a far cry from the bits and pieces we had to dig out in early '77 and '78. Many key pieces of information came to us via PET USER Notes, PET Gazette and The PAPER. Of the three, only The PAPER remains! We have an excellent 'slick' magazine, COMPUTE, which has replaced the other two, and it is a valuable asset; however, it's nice to have a folksy newsletter. In The PAPER we can ask 'dumb questions' and share initial findings without having to wait until they're finely polished.

One of the most valuable parts of The PAPER is the Reader I/O. A word to the reader NWOR and others who are getting started. If you really are a beginner, then machine language programming is NOT the place to begin. It's like eating an apple (no pun intended) by starting with the core! Perhaps it can be done, but it will be much easier (and taste better) if you start at the outside and work your way in. Learn BASIC, do a LOT of programming, and get a feel for what the computer can do. Then when you're ready for machine language, you'll be able to see its great strengths and many terrible weaknesses. For some purposes, it's worth all the trouble it takes. For many other cases, it's a disaster and should be avoided. The 'Machine Language' tape provided by Commodore is a monitor, which allows you to inspect and change values in absolute memory locations. You must know the architecture of the 6502 computer (its registers, data paths and instructions); understand the relationships among binary, hexadecimal and decimal number systems; and keep track of addresses and pointers; all of which you can affect and manipulate with the monitor. Of course, if you make a mistake, you may lose control and will have to start over. Many of 'these' crashes can be recovered by using UN-CRASHER from International Technical Systems, PO Box 264, Woodbridge, VA 22193. It costs \$14.95 and allows you to set your BASIC or machine language program back in most cases.

Annie Lee's description of the PET's floating point arithmetic routines was very close to what I would like to see for all ROM routines. Explanations should include where the routine is located, where the input should be placed before calling the routine, and where the result is left when it's done. Am I correct in assuming that the locations given are for the old (Version 1) PETs? Annie is the author of a Machine Language Guide for the PET with locations of routines for old and new PETs and numerous examples of how to use them. The Guide is available from ABACUS Software, PO Box 7211, Grand Rapids, MI 49510.

I take issue with Francis Turco's defense of PIMS. A faulty program is not overwhelming unless you are overwhelmed. Any company or individual offering software for sale has an obligation to ensure that it works correctly under all reasonably predictable circumstances. If it fails to do so, the seller has an obligation to fix it, or to refund the customer's money. Someone who buys a program should not have to debug it! On the other hand, people who fail to pay for software should get only copies that always fail. Quality design and

responsible maintenance and service are good reasons to pay for software.

I did enjoy Fran's article and program on the 'Better List'. The program appears to well-constructed, and is certainly well-commented (many good REM statements). Not only is it a good model for new programmer's to study, but thorough study of the program will lead to a better understanding of the way BASIC programs are stored in the PET. The 'key words' in lines 63988-63995 are available in ROM locations in old or new PETs, although with old PETs, it takes a tiny machine language routine to be able to retrieve them. On the new machines, the following 'command mode' will display them on the screen: FOR I=49298 TO 49551: ?CHR\$(PEEK(I));:NEXT

The corrections in the March/April issue to James McArthur's excellent SEARCH program were welcome. I had been disappointed when I typed it in and it didn't run. After making the correction specified (hex location \$0375 should have contained \$38 rather than \$28), I found that the program still didn't work. With the help of my son, Rick, I found that I had left out 8 bytes, which meant that my listings on page 35 and 36 were incorrect. I'll be glad to supply corrected listing to anyone who asks.

For those of us who want to learn assembly language, the listings on page 20 and 21 of the March/April issue were excellent. Listings should show mnemonic or codes, symbolic operands, lots of comments and the actual hex code that goes into each memory location. Incidentally, that listing was produced by Carl Moser's Eastern Software House Assembler. I have no connection with that company but recommend its products wholeheartedly. There is a tape version, as well as, a more extensive one for disk owners. For anyone who wants to get really into machine code a good, high-speed assembler is a necessity.

Again, thanks to Fran Turco for the excellent summary of 6502 op codes. This should make raw machine code more understandable to many readers. Of course, I'm never satisfied...it would also be nice to have a 'decode' list, where the hex values are in numerical order. This way we code look up a value and see what op code it represents. Also, it would be nice to have a diagram showing the various registers and the paths along which data can flow in the 6502.

On page 17 of the March/April issue, the Software Maintenance bulletin should have referred to a Micro Software System product (PO Box 1442, Woodbridge, VA 22193). If you like the idea of these bulletins, tell your dealers and write to other manufacturers.

Kudos to Paul Sparks for his work on C.L. Buchanan's screen print program, and an excellent article. Mr. Buchanan is a treasure trove of worthwhile information, and has been a great help to many of us. One of his major areas of interest is in the tape I/O subsystem. If any of you have made interesting discoveries in that area, please send them to him at 5000 Shorton Dr., Camp Springs, MD 20031. Add a note that he should put it all in an article for us.

While we're on the subject, how about more articles from previously unpublished authors? (That means YOU!)

Ray Davidson, in MEM EXPLORER AGAIN, demonstrated how one idea can be explored from several different angles and can provide new learning experiences. It should be noted that the commercial version offered by Micro Software Systems includes a routine to merge it with your program.

It's been a good day but I'm bushed. Goodnight and see you soon!

Meddling with Middle Strings

by Gerry Eisner

The MID\$ function is a useful tool in graphics and motion graphics, because interesting results can be achieved with relatively little memory. MID\$ dissects a string in such a way that the string elements may be used separately or in groups. In MID\$(A\$,X,Y) the computer selects from the string A\$, Y number of letters starting at the Xth letter. For example, ? MID\$("ABCDE",3,2) will output CD.

Let's start with a spinning stick: (NOTE - Due to the limitations of printing many characters cannot be truly represented. BE CAREFUL.)

```
10 A$="@NBM":REM ALL THESE SHOULD BE SHIFTED
20 FOR X=1TO4
30 ? MID$(A$,X,1)"(c1)": REM PRINT EACH IN ORDER THEN CURSOR LEFT
40 FOR I=1TO10:NEXT I,X:REM TIME DELAY TO SLOW MOTION
50 GOTO10:REM REPEAT
```

To reverse the spin change line 20 to:

```
20 FOR X=4TO1STEP-1
```

You may want to try juggling two spinning sticks by adding more MID\$ functions and setting the space between them with cursor moves.

For a different graphics effect, keep lines 30, 40, and 50 and add:

```
10 A$="$RF@CDE#EDC@FR":REM ALL THESE SHOULD BE SHIFTED
20 FOR X=1TO14
```

We have just seen examples of varying the starting point of the MID\$ function. Other effects can be achieved by changing the length of the string elements used while keeping the starting point constant. Try:

```
10 A$="CHARLIE BROWN"
20 L=LEN(A$):REM NUMBER OF LETTERS IN THE STRING
30 FOR Y=L TO 0 STEP -1
40 ? " "MID$(A$,1,Y):REM THE SPACE ERASES
50 FORJ=1TOY:?"(c1)":NEXTJ:REM MOVE STARTING POINT TO COVER OLD STRING
60 FOR K=1TO75:NEXTK,Y
```

Another exercise would be to make the name appear letter by letter on a blank screen. If the name slides in from right to left, you would be maintaining the same starting letter. However, if CHARLIE BROWN slides in from left to right, both the starting letter and length must vary at the same time:

```
10 A$="CHARLIE BROWN"
20 L=LEN(A$):Y=1
30 FOR X=L TO 1 STEP -1
40 ?MID$(A$,X,Y)
50 FOR J=1TOY:?"(c1)":NEXTJ:Y=Y+1
60 FOR K=1TO75:NEXTK,X
```

A much more difficult 'final exam' would be to make CHARLIE BROWN slither down a flight of stairs and off onto a landing using two middle string functions working together.

By varying the selection of X randomly, a form of a squiggle can be displayed:

```
10 X=INT(RND(1)*4)+1
20 FOR I=1TO10*RND(1)
30 ? MID$(" (cl)(cd)(cr)(cu)",X,1)*"(c1)":REM CURSOR MOVES
40 NEXT I:GOTO10
```

If this solution bounces off the top of your screen too much, add a couple more (cd) to the string and change the 4 in line 10 to a 6. Also try changing the 1 in the MID\$ function to a 2.

By now your imagination should be stirred about the intriguing possibilities in the use of MID\$. You can create false cursors or spinning symbols in the window of a slot machine. Have fun!

(Pub. note: I guess I passed Gerry's final exam. I wrote a program which causes CHARLIE BROWN to come out of a sliding door at the top of a flight of stairs. These letters then march down the stairs, step over each other and line up on the lower landings.)

SEARCH Note

by James F. McArthur

I have gotten some inquiries about how to modify SEARCH for the Version 2 PETs. This probably won't be very hard to do, except that I don't have one to experiment with, although I do have considerable information for the new system.

The present version of SEARCH uses part of the BASIC input buffer, which is in page zero on the Version 1 PETs. This area is fairly safe to use, most of the time. Unfortunately, among the many changes Commodore has made in designing Version 2 is the relocation of this buffer to other memory locations. This means there is very little page zero memory to play with.

To modify SEARCH for the new system, I will need information on what areas in page zero can be used without making the PET so crazy. Specifically, I need a block of four adjacent bytes and three single bytes, for a total of nine bytes. These bytes must not be affected by floating point accumulator operations. Perhaps someone has this information and could pass it along.

In the meantime, here is a modification to make it possible to stop SEARCH with the STOP key. At the time that I sent SEARCH to The PAPER, I didn't know how to squeeze this feature into the 198 bytes available. Since then I have found a way to fit it in by rearranging a few things. With this modification, SEARCH is 196 bytes long.

Step 1: Retype line 130 as shown:

```
130 V$(1)="20E1FFA003A200B11A991C0095B1E88810F5A51C051DF09"
```

Step 2: Delete first 16 characters in quotes in line 140.

If you have suggestions or question please let me know at: 3240 Blundell Rd., Falls Church, VA 22042

A Better Screen Copy

by J. R. Bromley

The October 1979 issue of The PAPER contained a screen dump subroutine for use with the Commodore printers. The program had a problem in that it would not respond to anything within quote marks. Carl Hildon of Commodore Canada gave me an idea for a fast(er) screen dump routine which I will call screen copy. The routine I have developed does not pose a problem with quote marks; is somewhat faster than other published routines and gives the user visual indication that a 'copy - no copy' decision is required.

The program listed is written for the old ROM PETs, however, Table 1 gives the modifications necessary to use it with the upgrade ROMs. It uses the concept that the screen is a logical device and may be opened and accessed in the same manner as a data file. Each character on the screen is read through the use of the GET# command. The GET# command 'gets' the character displayed on the screen in ASCII format. There is, therefore, no need for any conversion to ASCII within the subroutine. This makes the subroutine faster than others.

The subroutine faithfully reproduces each character on the screen (recognizing that reproduction of graphics characters requires a compatible printer). However, the screen image itself may scroll up as the characters on the last two lines are read. This is not usually a problem since the image is on hard copy. One feature is that a visual indication that the subroutine has been called is given. This is accomplished by the use of a simulated cursor at the lower right hand corner of the screen. This is flashed to attract your attention and allow you to make a copy-no copy decision. If you strike a 'P' the screen copy routine is executed. Any other key will allow the program to resume execution at the point where the subroutine was called. This permits the user to insert the subroutine call anywhere in the program and ensure that the program will not progress past the desired point while at the same time providing a positive indication that the desired point in the program has been reached. Once the 'copy-no copy' decision has been made the character originally occupying the lower right hand corner is replaced and the flashing cursor is removed.

My printer is not a Commodore model and does not have graphics capability. For this reason I have included the line of code at 63690. This code examines each character and prints a space for each graphics character found. This maintains relativity of the printed copy although the attractive graphics you may have spent hours to develop for the screen will be lost. If, in line 63960, you replace (...THEN GG=CHR(32)) with (...GG#=CHR\$(42)) an asterisk will be printed instead of a space. If you have a printer capable of reproducing PET graphics, however, you should delete line 63960.

Because the purpose of the coding may not be obvious to everyone, I have included the following description of the purposes of selected lines of the subroutine.

<u>Line No.</u>	<u>Description</u>
63900	Opens the screen (device 3) and printer (device 4) as logical files 1 and 5 respectively.
63905	Obtains the PEEK value of the character existing in the lower right hand corner of the screen.
63910	Establishes the position of the flashing cursor as the lower right hand corner of the screen.

63915 Turns the flashing cursor on.
63930 Turns the flashing cursor off and replaces the original character lower right hand corner.
63945 Obtains the length of the line being printed. This will be 39 if the line contains less than 40 characters, else it is 79.
63950 Uses the line length obtained to set the number of iterations for the inner (JJ) loop. That is to say the number of characters to GET for a line on the screen.
63955 GETs the next character from the screen.
63960 Exchanges graphics characters (for printers not capable of PET graphics) for a space. Omit this if your printer is compatible with PET graphics.
63970 Closes the inner loop and forces a carriage return at the end of each line. Begins examining the next line (next iteration of outer (II) loop).

Listing

```

1      REM SCREEN COPY BY J.R. BROMLEY
2      REM JANUARY 1980
63900 OPEN 1,3: OPEN 5,4
63905 CH=PEEK(33767)
63910 POKE 224,192: POKE 225,131: POKE226,39
63915 POKE548,0
63920 GET GG$: IF GG$="" THEN 63920
63925 IF GG$<>"P" THEN POKE 33767,CH: GOTO 63975
63930 POKE 548,1: POKE 33767,CH
63935 PRINT "(home)": REM CURSOR HOME
63940 FOR II = 0 TO 24
63945 LL = PEEK (242)
63950 FOR JJ = 0 TO LL
63955 GET#1,GG$
63958 GG = ASC(GG$)
63960 IF GG<32 OR (GG>127 AND GG<193) OR GG>218 THEN GG$ = CHR$(32)
63965 PRINT#5,GG$:
63970 NEXT JJ: PRINT CHR$(13): NEXT II
63975 CLOSE1: CLOSE5
63980 RETURN

```

Table 1

<u>Line No</u>	<u>Original ROM</u>	<u>Upgrade ROM</u>
63910	224	196
63910	225	197
63910	226	198
63915	548	167
63945	242	213

Location 33767 is the same for both versions.

Cassette Survival Hints

by D. Sherwood

I use cassette tapes rather than floppy disks for two reasons:

- 1) Poverty: I can live with the longer time locating and loading when I compare the price of a disk drive with a cassette.
- 2) Rationalization: Disk drives are a passing fad.

These excuses will change if and when my finances improve, in the meantime I would like to make the use of tapes easier and more reliable. These hints are a collection from many sources and my own experiences.

TAPE QUALITY

This is a combination of good tape in a sturdy housing, no five for \$1.00 specials. For master program tapes, C-30 and C-60 lengths seem the best. C-90 are not really necessary and usually have thinner tape which stretches more easily. TDK brand has served me well.

For programs in regular use or as current data files, C-10's are the answer and they can easily be purchased of good quality and at a reasonable price. The shorter tape allows easy access to both sides for more economical use of tape. Computer Way, Madison WI supplies C-10 AGFA tapes in a good housing at about \$1.00 each. I have 50 or so in constant use and have had no problems.

TAPE STORAGE

Always store tapes in a box, hard or soft, and then in some sort of storage rack. Never touch the tape, rewinding will keep scrubby fingers off the tape and will leave only the leader exposed. On master program tapes and those used to store tape files make sure to knock out the record-protect tab on the back of the cassette. If you have to add to the tape later a piece of transparent tape over the hole will do the trick. Meanwhile this will prevent the not so magical disappearance of programs. If reusing tapes for data files, I suggest the use of a bulk eraser. This will ensure you do not read unwanted data from the original tape.

CASSETTE HEADS

Clean and demagnetize your cassette deck tape heads regularly. This alone will reduce problems. The frequency of cleaning will depend on hours in use, but weekly housekeeping on the tape heads is well worth the time. A de-magnetizer for about \$10.00, a low residue cleaner, and a box of Q-tips is all that is needed.

The suggestions above may seem commonplace and not worth repeating, but if you have agonized over a scrambled program due to bad tape or been unable to establish whether the reading of faulty data is the program or tape, you know these steps are worth the time. The final suggestion presented here is offered with no assurances beyond the fact that I have used it with no ill affects over the past year. This suggestion involves a modification of a second cassette deck which will probably void your warranty. I report here what it did for me and if the idea is not original then it is a case of reinvention not borrowing the idea.

TAPE HEAD ALIGNMENT

Tape head alignment when done by your dealer with his alignment tape is accurate and should cause no problems. When using tapes from other machines of dubious alignment or some tapes not in use for a long time problems may occur. One commercially prepared tape of twenty utility programs would not load three of the programs easily. These programs seemed to be of slightly different alignment. Attempting to solve this problem led to fitting a mini-jack to a second cassette deck and using an amplifier and speaker as an alignment aid. This may sound like overkill, but I can now read most valid tapes. I can also be sure that my tapes will load on my friend's machines. This is particularly important when exchanging tapes through the mail. This suggestion should only be implemented if you have two cassette decks and the first remains aligned to your dealer's standard. (Note: Parts numbers given are Radio Shack's.)

Remove the cassette deck from its case. In the upper part of the case there is a small hole covered by the white metal strip that labels the switches on the cassette. Carefully drill through the strip with a small bit and enlarge the hole with a small file. Make the hole large enough to accept a small screwdriver. This hole will allow access to the adjustment screw when the play button is depressed.

In the lower part of the case, fit a normally open mini-jack (RS #274-297) where it will be out of the way; the upper right is suggested. To wire the mini-jack, place the cassette with the buttons toward you. The common or ground lead should be connected to the upper left of the PC board. There are two leads from the underside, yellow and black. The other lead should be connected to the lower right of the PC board at the third solder joint from the bottom of the board. It is the junction of two resistors on the underside. Now dress the leads out of the way and reassemble the cassette player. Connect the deck to the PET and LOAD a program as a check. After it is LOADED type `PRINT PEEK(630),ST` for old PETs and `PRINT PEEK(192),ST` for the new PETs and then `RETURN`. In both cases you should get `0 0`.

Now connect an amplifier and speaker (RS #277-1008) to the cassette, and when the PLAY button is depressed you can hear the tape; this allows you to carefully adjust for maximum clear sound. You will hear a steady tone, followed by data, and then another steady tone and then program data. If you are LOADING a program, it should report `FOUND` and `LOADING` by the time you hear the start of the second steady tone. In most cases it will require only a slight adjustment to align to another machine. When you have finished you could realize to a tape recorded on the first cassette deck.

If you try this modification I would like to repeat the following suggestions.

1. Keep the first tape deck aligned to the correct standard as set by your dealer, and use it to write your master files.
2. If you have changed the alignment on the second deck, you should realize it using a tape recorded on the first deck.

These suggestions should help avoid writing a tape with varying alignment. This, of course, was the problem you were trying to solve in the first place. I would welcome any suggestions or comments via The PAPER or to me directly at PO Box 2211, Dartmouth, N.S. B2W 3Y2

ASSM/TED

A Product Review

by Jim Fowler

Type: Software

Model PET: all models without disk and with 16K RAM

Source: Eastern House Software

3239 Linda Dr

Winston-Salem, NC 27106

Price: about \$80 depending on options bought

The ASSM/TED is an assembler and text editor package available for old or new ROM PETs and is written entirely in 6502 machine language. A Graphics Compiler is available as an assembly program to be loaded by ASSM/TED. An enhancement which comes with the compiler reduces the interference between PET BASIC and ASSM/TED in page zero.

ASSM/TED documentation is very complete and well organized so the beginner can go from start to finish with a PET to try it out. Be prepared to spend some time unless you are a professional programmer who already knows what to expect from assemblers and editors. You will need at least 16K of RAM to use the program although the more the better. TED could be used by itself as a text editor. It works by numbering the lines of text. To change a line you can use the PET's screen editor but the order of lines, copying or moving of lines is done using TED commands. When the text is output the line numbers can be suppressed for assembly programs. TED is nice but its ASSM, the assembler, that really makes the package worth the price.

ASSM uses all the standard 65XX mnemonics and formats. In addition, it has MACRO capability or the ability to write a sequence of code and then call it by name. Over 20 pseudo-ops are available. When assembling you may continue even after errors occur. Thirty six different errors are detected and the assembler displays them as they occur. The enhancement available with the graphics compiler also allows you to incorporate macros with ASSM/TED and also incorporates a library of labels with their addresses. This allows you to use your favorite PET operating system routines. All this is put into memory so that you need only type it once.

Once you have composed your assembly program with TED you can save any length of text to tape or disk. The Asssembler will then get the text from memory, tape or disk. Once the assembler is done you save the assembled program using the PET's TIM MONITOR. You can use one address to store the machine language but force the assembler to use another base address for the code. This allows you to put it anywhere you want and move it later to a more permanent address. Finally, if you have two tape decks you can write moveable code. A listing of a RELOCATING LOADER is provided with ASSM/TED. This program loads the machine language wherever you want it. It takes the code, calculates the true addresses based on that location and puts it in memory.

The Graphics Compiler is an assembly language text program for ASSM/TED to assemble. You write your graphics program in terms of the compiler's functions. These are: straight lines at angles you specify composed of symbols you specify for lengths you give, printing of strings of symbols, getting input from the keyboard, various conditional branches and several ways of setting the values of variables. All this is done with symbols in place of numbers. You can move the cursor all over the screen in either absolute or relative terms (by giving row and column, not true address). This can be interspersed with timed delays. Some game writers may find this

interesting as it stands. My own view is that the compiler is useful mainly as an example of an assembly program that allows you to invent your own language and compile it so that you end up with a program in machine language.

These programs are not expensive if you figure the time they will save the serious programmer. They could increase your output ten to fifty times and reduce your aggravation by the same amount. They're very high-powered and professional. The programs take more memory than some may have but are worth every penny for those who wish to step up to a more powerful and efficient programming level.

(Pub. Note - A more powerful and more expensive disk based version of this package is now available. Watch for a complete review of this package in the next issue.)

Now You See It...

by RHB

You try listing the program you just bought and as you do lines appear without any line numbers! They just hang there in mid-program! Other lines have numbers but nothing following them. Actually, this technique is quite simple. One reason for using it is that it is a nice way to add your name and other information to your program. This really can impress your friends especially those who don't read *The PAPER*. Secondly, you can hide some important information in these lines if you don't want your program changed by others easily.

To perform this miracle follow the directions below exactly.

- 1) Type: 100 REM:?"
- 2) Now hit the DELEte key to delete the second quote: 100 REM:?"
- 3) Hit the INSerT key 10 times; then hit DELEte key 10 times
- 4) Now type your name or another message.
- 5) Now list just this line.

Your name should now be hanging in mid-air without a line number. To make a line which shows nothing at all in a list you can follow the above procedure but do not add any name or message. If you make a line 110 like this your name in line 100 will be separated from the rest of the program by a blank line.

Now you can experiment with different line numbers, various messages and hidden information. Remember however, since the line number and much information disappears you cannot make corrections once the line you typed in leaves the screen. Type these special line one at a time and then list them to see what they look like. If there is a mistake you can correct it in the original line you typed.

To make some important information nearly invisible is fairly simple. Before the REM in a disappearing line place the important program information. You might want to set an important variable to some value; W=100. For each character in this information hit an extra INSerT and an extra DELEte in the procedure described.

This procedure is far from foolproof. People who are looking for it may be able to read the information especially if the line is long. A listing to a printer will also show the entire line. If you have problems make sure you followed the directions and then contact me via *The PAPER*.

Auto Repeat Keys, Version Two

by Roy Busdiecker

Way back in Issue 10, Volume 1 (Dec '78) of The PAPER, there appeared a brief article entitled "Automatic Repeating Keys" submitted by The Software Shoppe. The article is on page 27, for those of you who save valuable relics!

The program was a gem...the fact that repeating keys is desirable is attested to by the fact that Commodore built a version of that feature into the 8000 series machines. Unfortunately, I felt that the article left something to be desired. It provided a program that worked, but no explanation as to HOW it worked. Furthermore, it worked only on the old Version 1 PETs with the small keyboard.

Wayne Sung to the rescue! In Issue 3, Volume 2 (August '79), Wayne described changes which would make the program work on the Version 2 PETs (which Commodore now calls BASIC 3.0), and gave some narrative explanation about what makes the program tick. However, from my point of view, Wayne stopped a little short.

To make the machine language program understandable, several items are very helpful, if not absolutely essential.

1. The Code (the numerical contents of specific memory locations, to which control is transferred to accomplish the task). The original article did this in the most minimal manner through a BASIC program and a combination of READ, DATA, and POKE statements that did more to hide the way it worked than reveal it. Wayne did better by talking about operations codes and changes in the program, but didn't tie it down by showing the whole program together.
2. The operations (very closely related to the Code...statements like ADD, STO, LDA, etc., that have meaning to the Assembly Language Programmer, and are described for our computers in the MOS Technology 6502 Programming Manual and a number of later books.) If you're using an assembler (and the best I've seen for the CBM products are those by Eastern House Software), then you'll recognize this as the assembly language listing you put in, and the fact that the assembler gives it back to you with the machine language equivalents.
3. A narrative description, preferably keyed to the assembly listing. This may cover design and logic information, but must at least provide operating instruction and the parameters whose values may be changed to alter the performance of the program.

This article provides an example of what I mean, although the narrative it provides is most limited (no use beating a dead horse!).

THE PROGRAM: The program described herein is another variation which works on Version 2 PET/CBM 2001 series computers of any memory size. It will work with either tape or disk storage, but will cause problems if another program is entered from tape after this one is loaded, since this AUTO REPEAT KEYS program uses parts of both the first and second cassette buffers.

THE CODE: Figure 1 shows the values which must be entered to make the program work. Values are shown as they should be entered using the built in TIM monitor on version 2 machines.

THE OPERATIONS: Figure 2 shows a disassembly listing. The main difference between this and the assembly listing is that the former uses no names for variables or labels, and in fact is produced backwards by looking at the code and figuring out what would have to be fed to an assembler to make the CODE come out. That's an oversimplified explanation.

THE NARRATIVE: Remember, I said it would be brief!

1. To turn the automatic repeating feature ON or OFF enter SYS802.
2. To change the DELAY before repeating occurs, POKE 827,X. X is preset at 20. A higher value delays longer. If you make the change while in the monitor, the hexadecimal equivalent of 845 is 034D.
3. To change the speed with which the repeating character zips across the screen, POKE 845,X. X is preset at 3. A higher value slows it down. If you make the change while in the monitor, the hex equivalent of 845 is 034D.

See if this handy routine doesn't make your computer easier to use.

```

M:0322-035C   0  1  2  3  4  5  6  7
M:0322-0329  78 38 A9 E9 ED 91 00 8D
M:032A-0331  91 00 58 60 AD 97 00 C9
M:0332-0339  FF D0 0C A9 00 8D 20 03
M:033A-0341  A9 14 8D 21 03 D0 19 EE
M:0342-0349  20 03 AD 21 03 CD 20 03
M:034A-0351  B0 0E A9 03 8D 21 03 A2
M:0352-0359  FF 8E 97 00 E8 8E 20 03
M:035A-0361  4C 2E E6 FF 00 FF 00 FF

```

```

D:0322-0356   1  2  3   MNC-CODE
I:0322         78         SEI
I:0323         38         SEC
I:0324         A9 E9      LDA #$E9
I:0326         ED 91 00   SBC $0091
I:0329         8D 91 00   STA $0091
I:032C         58         CLI
I:032D         60         RTS
I:032E         AD 97 00   LDA $0097
I:0331         C9 FF      CMP #$FF
I:0333         D0 0C      BNE $0341
I:0335         A9 00      LDA #$00
I:0337         8D 20 03   STA $0320
I:033A         A9 14      LDA #$14
I:033C         8D 21 03   STA $0321
I:033F         D0 19      BNE $035A
I:0341         EE 20 03   INC $0320
I:0344         AD 21 03   LDA $0321
I:0347         CD 20 03   CMP $0320
I:034A         B0 0E      BCS $035A
I:034C         A9 03      LDA #$03
I:034E         8D 21 03   STA $0321
I:0351         A2 FF      LDX #$FF
I:0353         8E 97 00   STX $0097
I:0356         E8         INX
I:0357         8E 20 03   STX $0320
I:035A         4C 2E E6   JMP $E62E
X:END

```

Program Exchange

by Charlotte Deschamps

The Exchange has finally begun to operate successfully after a few complications. We have updated the list of programs available in an attempt to offer you better quality programs. This issue contains a list of all of the programs in the Exchange with descriptions. Updated lists of programs available for exchange can be obtained by sending 50 cents and a self addressed envelope to the address listed below. Please allow 3 weeks for delivery.

The Exchange is meant to distribute original and noncopyrighted material to subscribers of The Paper for a nominal fee. Copyrighted programs may not be submitted and other people's programs may not be submitted without their approval.

Programs may be submitted on TAPE or DISK. Disks are preferable as tapes take longer for us to process. The exchange is intended to be 'one for one' and no provisions have been made for people who have no programs to exchange.

In order to insure efficiency all programs will be sent out to participants on the same date each month. Programs recieved on or before the 1st of each month will be sent out on the 21st of that month. For example, if your program is received on or before December 1, 1980, exchange programs will be mailed back to you on December 21, 1980. If your program is received on December 10, 1980, exchange programs will be mailed back to you on January 21, 1981. This delay is to insure that programs received are properly reviewed and also allows you to predict when you will be receiving your exchange copies.

The following procedure will be used:

- 1) Send your programs on TAPE or DISK to:

Program Exchange
c/o Charlotte Deschamps
1 Pine Place
Lindenhurst, NY 11757

Include a brief description of each program and its source. Include a list of specific programs you want from the list and \$1.00 per program.

- 2) Within three days after your programs are received you will be sent a postcard stating the date received and any other pertinent information.
- 3) Your tape or disk will be returned to you with your requests and our Exchange List. If we feel people would be interested in your program and it does not duplicate any program on the Exchange, it will be included in our list. You will receive one program for each non-copyrighted program you send us.

If you see any copyrighted programs on our list please let us know immediately so that they can be removed. Many programs list 'unknown' as author since no name was included within the program. If you recognize a program, please send us the author's name. If you have any suggestions to improve this process please let us know. Input is welcome.

Program Exchange

Educational

Op Drill	J Comito	simple drill with 4 basic operations student picks one number
Balloon	unknown	through a dart to hit a balloon, estimate decimal fractions on number line
Punct	Furry & Ansel	punctuation drill, easy to change problems in DATA statements
Voc	L Braun	hangman-like game with definition of word included
Dart Est	unknown	difficult math problems require accurate estimates within time limit
Primer	unknown	deduce numbers in a multiplication problem
Who	unknown	ask set questions to uncover identity of biblical figure, adaptable for many situations
Assembly	J Woelfel	simple introduction to idea of assembly language
Points	J Comito	drill in how to plot points on coordinate plane
Slope	J Comito	lab exercise to teach slope form of equations $y=mx+b$
Equations	J Comito	drill to test slope form, $y=mx+b$
Plot	J Comito	you give PET equations to plot, it gives slope and plots
Trip Cost	unknown	estimate trip cost given time costs and number of days
Sieve	unknown	graphically finds primes from 1 to 1000
Knight	R Bressler	graphically illustrates Knight's tour of a chess board
Marbles	M Oselsby	drops marbles to illustrate distribution according to laws of probability
Fraction	unknown	shoot arrows at target estimate fractions
Morse	H Sherman	morse code practice
Oregon	R Bressler	travel along the Oregon Trail, balance food and other needs, no graphics, instructions called Pioneer
Kingdom	R Bressler	manage your lands well to maximize profit and minimize revolt
Civil	R Bressler	Fight the famous war as the Confederacy, decide distribution of money and your strategy
Jury	R Bressler	choose a 12 person jury given legal observations, NOT GUILTY
Stocks	unknown	buy and sell to make your fortune
Elect 1	Huntington II	simulates elections of 19th century, you determine points for image, issues and party affiliation
Elect 2	Huntington II	simulates the 20th century
Water	unknown	manage water supply of a small town for agriculture, industry and home use
Hammurabi	unknown	can you do better than the original?
Epidemic	unknown	simulates disease in population
Life 64 x 64	F Covitz	Game of Life, old PETs only
Superlife	Bartonsmith	another version of Life for old PETs

Function Machine	P Rowe	you put in a number, it spits one out
Pakiana	unknown	discover the RULE
Matches	V Mills	program the dancer to perform in the way you direct
Fraction Target	B Finzer	take as many matches as you want but don't take the last one
Guess Fraction	B Finzer	drop bombs to hit a fraction on a number line
Blocks	unknown	guess the secret fraction on the number line, zooms in as you get closer
Basels	V Mills	discover the attributes of sets
		guess the 3 digit number sequence PET has chosen

Games

Hangman	C Sommers	traditional game with great graphics, several categories, negative reward
Pop Shop	unknown	good target game with sound
Tank Wars	unknown	two player action game controlled from keyboard, hunt and destroy enemy tank
Snake	P Rowe	manuver your ever lengthening snake to surround the opponent
Change 9	R Bressler	reorder a random arrangement of the digits 1 to 9
Chase	unknown	evade the clumsy robots and live to fight again, robots are stupid
Star Wars	R Bressler	fly down a narrow canyon, evade the enemy fighters and release your missile
Trapper	R Nathanson	trap the wolf with your three hunters before he sets to the sheep
Lunar	unknown	fire your retro rockets to land the lunar lander
Yahtzee	P Rowe	the popular dice game for one, good graphics
Osero	unknown	one version of Othello
Othello	JK Johnson	plays a pretty good game, has been speeded up with some advanced play
Noose	J Jesse	a hangman version where the PET or person picks the word, OK graphics
Breakout	JK Johnson	FAST version of the game with some machine language routines, old PET
Weigh	unknown	logic game to determine the relative weight of some unknown objects
Baseball	R Bressler	presents random number 'pitches' you must 'hit', no graphics
Simon	unknown	repeat a random light and musical pattern as long as you can
Deflect	unknown	place slashes to control path of ball, hit the targets, difficult
Wrap Trap	unknown	another version of Snake, Snake may travel off screen and back on
Copscat	unknown	repeat a lengthening series of card symbols
Klinson	unknown	surround the enemy with shots as he tries to escape, don't hit him
Derby	J Baker	good graphic horse race

Concentrate	unknown	turn over the 52 cards of the deck, try to make as many matches in as few turns as possible
Gallery	F Weiler	shoot down PET graphics symbols as they move above your gun
Slot M	M Richter	excellent slot machine simulation, wheels 'turn', coins come out
Draw Poker	E Wuchter	you against the house, good graphics
Space Shooter	D Howe	line up the spaceship in your sights, fire your lasers, several options
Starlanes	unknown	build an empire by capturing adjoining stars, board game
3DTTT	unknown	3 dimensional tic-tac-toe, slow but with good graphics
Watson	S Irving	a Mastermind-like game using crimes and suspects
Amazin	H Koch	creates a random maze on the screen, times your escape
5 Draw Zone X	D Howe R Bressler	play against the odds, shows odds try to find the intersection of four colored zones
Press Ups	R Bressler	two players try to capture as many bees as possible
Token	unknown	underground classic, try to empty a waterpipe without choking, graphics
AFO	unknown	release balloons to down a UFO, sit back and see what happens
Invaders	unknown	popular arcade game written in machine language, difficult with superb graphics, specify your machine
1000 Miles	unknown	version of Mille Borne, card game with travel motif, first to score 1000 wins, play against the PET
Solitaire	unknown	plays several version of Patience, ok graphics, all cards displayed
Robot	C Sommers	evade the robots or fry them on the fences, difficulty adjust to your level
Dominoes	M Richter	good graphics in simulation of standard game of dominoes
Sink	M Richter	very good game of Battleship, graphics, Pet plays surprisinally well
Subs	unknown	release depth charges to sink subs below, good graphics
Dam Buster	E Williams	release bombs from different altitudes to break the dam, good graphics
Spades	R Revis	good version of the game against the PET, good graphics, PET hard to beat
Termite	H Sherman	random termite eats its way around board, player slobbed wins
Emporer	Sphinx	manage your kingdom to make a profit, don't make the peasant mad
Dart Board	unknown	throw darts with one of three motions
Sandpile	M Oselsby	use your bulldozer to level a sandpile
Pinball	B Jaeger	slow version of the game, fairly easy
Spinners	D Ford	advance around the board, two players, takes some daring to win
Pyramid	S Horowitz	a good version of the solitaire game

Useful

Metrics	R Bressler	a simple program that converts a person's weight and height to metrics
Biotimer	R Harrison	calculates and displays biorhythms
Titler	S Irving	displays 4 line of LARGE characters from user's input, good for video titling or other displays
Break Even	Commodore	analyze break even point for sales
Filehandler	based on fixed PET Gazette	and variable costs tape file handling program; read, write or add data to file; OLD ROMs
TIM	Commodore	machine language monitor for OLD ROMs
Hist	B Eisenberg	plots NICE histogram of random numbers
Renumber	B Honders	renumbers programs lines; OLD ROMs
Weekday	L Duggan	calculates day of week; 1900-2000
Bigtime	Commodore	displays time in large digits
Subplot	unknown	uses quarter squares to plot your function; function must be scaled
Framer	unknown	several options to surround your text with a border for highlighting
Dec Format	unknown	line up decimal points in \$ amounts
CMC Format	unknown	align decimal points or right justify numbers to printer
Mod Clock	Lindsay/RHB	analog clock face showing time and AM/PM; displays time in most major cities; set alarm or appointment
Tape Success	unknown	write and read data file tapes easily
Non-list	Lindsay	make lines disappear with this short subroutine ; will not list in printer
Stop Key	unknown	disable stop key; OLD ROMs
Routines #1	various	routines tested for old ROMs; includes line erase, line delete, line numberer
Routines #2	various	repeat key in BASIC and ML, find, outline, look at PET character, delete
Graphix	W Swan	routines tested for OLD ROMs; includes input, case converter, keyboard buffer
Autowriter	unknown	contents, ML screen filler, self listing program, upper/lower reverse
List Protect	unknown	excellent routines to help in graphic presentations; OLD ROMs
Typewriter	M Richter	automatically writes data lines into program, useful technique
React	D Haluza	place a special REM after each line, run the subroutine and the lines vanish
Digit Span	R Bressler	small word processor, works form tapes
Letter Span	S Irving	test your reaction time
NFL	E Burkhardt	remember and repeat sequences of random digits that get longer and longer
Sample Plots	unknown	repeat ever lengthening letter sequences, forward then reverse
Cursor	G Yob	predicts probable NFL scores, you must enter a LOT of data
Deal Sub	unknown	graphs of various functions displayed continuously on screen
		tired of the same old blinking box ?
		card dealing subroutine

Real Cursor	D Haluza	uses PET's real cursor during GET, two nice demonstration programs
Bases	unknown	convert between hex, decimal, octal and binary bases
Mach Plot	unknown	ML fast plot routine; OLD ROMs
MEM Explorer	unknown	displays 20 consecutive bytes of memory at a time; OLD ROMs
Sort Routines	unknown	includes quick, bubble and shell sort of randomly generated numbers
MPG	unknown	keeps mpg records for cars on tape
Disitab	unknown	shows different ways to print \$ amounts
Alphagrph	unknown	create and display pictures for each letter of the alphabet
Organ	J Butterfield	turns the PET keyboard into a musical keyboard, record and playback music
Random Music	unknown	plays randomly generated CB2 notes
Joelin	unknown	plays a Scott Joelin selection
Trans Music	unknown	plays transitional music which sounds like the notes belong together
Bach	unknown	displays notes which move up and down as it plays, enter your own notes
LSB Music	S Friedman	plays LSB music, has better tone than LSB, ML routine
Star Theme	BC Computing	plays theme from Star Wars
Superose	CC	draws nice designs based on polar coordinates, you input some info
Doodle	unknown	draws random pattern of squares
Patterns	unknown	choose various character patterns to create a symmetric design
Squiggle	Commodore	creates a random pattern
Scope	unknown	produces a kaleidoscope in BASIC
Monorail	unknown	build a train track and send the train on its way, add cars or increase speed
Spiral	unknown	random character spirals out from center
Cascades	RC	balls fall to fill spaces in a random 'maze', low spots fill first
Weather	CC	creates random patterns of rain, snow, and lightning
Gizmo	CC	creates random Indian rug patterns
Aphor	CC	generates random aphorisms from data
Anlist 1 & 2	unknown	make a program which does nothing when run but bounces a ball when listing

The programs here represent the original work of many authors. Some programs come directly from subscribers to The PAPER or members of LIPS. Some programs are from magazines and, in many cases, have been modified extensively for the PET. Other software has been obtained by exchanging programs with other program exchanges. Our intention is to distribute only those programs which the authors want us to distribute. Any program which an author does not want on the exchange will be immediately removed. To help expand the exchange we need new programs. The only way we can get new programs is from exchanges. If your friend wants a program that you got from our exchange, please ask your friend to write a program and send it to us. This is the only way we can grow.

Observations Resumed

by Roy Busdiecker

What a pleasure it was when I got the phone call from Ralph Bressler telling me that The PAPER was to resume publication! A big "Thank You" to all concerned. Many of us had really missed the availability of a publication inbetween the very small and local type of newsletter and the big, slick monthly or bi-monthly magazine. That area once filled by PET User notes (Gene Beals, now busy with AB Computers), PET Gazette (Len Lindsay), and The (PET) PAPER (Terry Laudereau), had slowly dwindled down to nothings. We were left without a person to person type of publication and it's good to have one back.

According to my records, the last issue of The PAPER that arrived before today (Nov. 3, 1980) was Issue 2/3, Volume 3, March/April 1980. Since Ralph said this was a double issue (#3) and Summer 1980 is written all over it, several facts must be deducted:

- (1) This must be Issue 4/5, Volume 3, and
- (2) Either
 - (a) We had a very late summer this year, or
 - (b) It took longer than anticipated to get this issue out (somehow, I suspect this alternative is more likely!).

Here's a quick update on what I've been doing for the last six to eight months. Micro Software Systems, which I founded in 1978 is now operating as a division of Virginia Micro Systems, Inc. In addition to software, we manufacture interfaces for printers and external video monitors, and operate a retail store in Woodbridge, VA. Our phone number is (703) 491-6502. I am no longer associated with International Technical Systems, although we carry and endorse their products. Finally, I remain a consultant for Edumatic, Inc. of Oceanport, NJ. Their Computer Forum in Red Bank, NJ, is an interesting combination of community educational facility and computing center, and retail outlet. If you live in New Jersey, it's worth visiting.

Now on to the Observations!

There is a slight inconsistency between pages 2 and 3. In the General Information blurb, we are advised that "all contents of The PAPER are copyrighted..." while the Editor's Notes on the next page say that "The PAPER is not copyrighted..." (we assume you meant "copyrighted"). Time for a policy huddle!

I'm fascinated by the name of The PAPER's new publisher...we'll now set the latest word, straight from the LIPS!

With respect to the Cross Referenced Memory Map, I have several comments. Everyone should be aware of the excellent material by Jim Butterfield that has been documented in several places. In addition to the original and "new" 2001 computers, we now need information on the 8000 series, too. Fortunately, many low-memory (0-1024) locations are used for the same purpose in the 8000 (and new-new BASIC 4.0 upgrades for 2001's) as in the popular "Version 2" BASIC 3.0 2001 machines. It would be helpful if all listings were given in both decimal and hex, as done in several other versions of the memory map.

In Doug's article on machine language, it was amusing to consider keeping time according to the number of dances (the BASIC example was supposed to take "over 16 minuets to run"). The point is well made.

however, and we'd only add that while machine language is generally much quicker to run, it's much slower to design and far, far slower to debug or modify. It's another tool which, when appropriate to the task, can be most valuable...but it would be a mistake to try to do all of your work in ML.

I enjoyed reading Bill Batcher's article on "The Evolution of a Puzzle". My entry, Bill, is to type ABELINCOLN*, where * stands for a shifted space (hold down the shift key while you press the space bar). It results in the output you required. I might not have thought of that so quickly, but I had just gone through a difficult program debugging exercise, where some inadvertently-shifted spaces fouled up the alphabetizing of a list.

In Doug's second article on machine language programming, he came very close to describing the operation of an assembler (a program that turns mnemonic instructions like LDA, STA, and so on, into the appropriate machine language codes). Assemblers make the generation of machine language programs much easier. Anyone who thinks he might need an assembler should be aware of the ones produced by Eastern House Software (their ad is on page 44 of the issue we're talking about). I have used and evaluated several of their high speed assemblers written in machine language, and they are the best I've seen.

The value of an assembler is illustrated in Jim Fowler's article, which (intentionally) avoids using it. The BASIC READ/POKE commands and strings of numbers is obviously very awkward. Slightly better, and satisfactory for very short routines is the built-in monitor on "new" 2001 series (SYS 64785) or 8000 series (SYS 54386). Nonetheless, for any machine language program of significant length, there is no substitute for a good assembler.

A command structure supported in some computers, but not Commodore's is the IF...THEN...ELSE... JoAnn's article ("Not Ifs Ands or Buts") reminded me of a technique originally brought to my attention by Bob Kaiser several years ago. The key fact that he pointed out is that if you use colons (:) to put multiple statements after a single line number, and the first statement is an IF statement, then ALL the statements following the IF will be executed ONLY when the IF-CONDITION is true (that caused some hard-to-find problems, back in the "early days" of 1977-78). Making the last statement in such a line a GOTO allows us to do the IF...THEN...ELSE...as in the example below.

```
10 INPUT A,B
20 IF A=B THEN C=10:D=12:E=C+D:GOTO40
30 C=9:D=11:E=C*D
40 PRINT E
50 END
```

If the inputs are equal, the program will print 22 as an answer, otherwise it will print 99. Try it!

Ralph, I disagree with the philosophy you present in "BASIC Does It Better". I agree it is best to do in BASIC what is done best in BASIC (which is most everything!), but just as there are some things done better in machine language, others are done best with POKE or SYS that depend on built-in tables or ROM routines. A good example is the stop key disable: as you point out, there is no way to do this in BASIC. A few points pertinent to the 8032 are shown below.

- a) Stop key disable: POKE 144,88. Re-enable with POKE 144,85
- b) Most low memory locations are the same as in the "new" 2001's

- c) PRINT PEEK(50003) returns a value of 160, which can be used to adapt programs automatically to BASIC 4.0
- d) The screen contains 2000 locations, rather than 1000 found on the 2001 series (incidentally, this screws up the displays of most "standard BASIC programs" as badly as it does ones which use POKEs, because programmers depended on the 40-column limit to dump the remainder of long strings on the following lines).
- e) To shift the 8032 screen to condensed graphics mode, hold BOTH shift keys down and press the "2" in the top row of the alphabetic keyboard.

There was one real gem in that article - I had never before seen the use of the "@" after a GOSUB or GOTO to introduce a REMark. A kudo to Ralph...I don't get many surprises of that sort these days.

JoAnn, I'd like to pick a nit with you! You've written a fine article on the design and implementation of a good program, then titled it "Writing ...". Good programming demands good design, and a precise arrangement of proper components. Writing can be imprecise and still be considered good. Let's describe prose as prose, and programs as systems, the two are more different than they are alike. By the way, I liked the article.

An addendum to the "un-authored" article entitled "Pretty Printer Listing": To set your listing in upper/lower case rather than graphics/upper case (and this will work only if you have a CBM 2022/2023 with "new" ROMs), use the routine shown below. I believe I got information necessary from Cursor, Goleta, CA.

```
OPEN4,4:OPEN7,4,7:PRINT#7,"A":CMD4:LIST
```

The article on Merging Programs contained some pitfalls. The POKEs given on page 41 were for the ancient (original) ROMs. If you have a newer machine, use the memory map on pages 4-6 to correct them. When I tried to use the technique shown on page 41 on my original PET, the FILE SAVE bus fouled things up, so it took some fiddling to get things to work right. MEM-EXPLORER and MICRO SET I from Micro Software Systems include those routines. If you only merge very short routines, you should be able to avoid the bus.

Although I'm used to seeing ads scattered throughout a magazine, I liked seeing them all together at the back of The PAPER. Unfortunately, inflation has taken its toll, and the ad from Micro Systems was an old one...Mem-Explorer, Cheque-Check, and Metric-Calc are now \$9.95. ALL of the programs are available for the 8000 series, and can be furnished on 2040-disk format for an additional \$5 (to cover the cost of the disk). New programs being finalized now include a Variable Dictionary and Cross-Reference (VAR-DICT/XREF) aid for programmers, a Disk Librarian with some neat features, and a Home Budget program tailored for either 8032 or 2001-32 (designed by Mike Johnson, with heavy shankshooting by yours truly).

This column has grown longer than usual..but with a double issue, I suppose that is to be expected. Please send in your comments, criticisms, or questions.

Overall, the first "new" issue of The PAPER looks good. Typesetting would be nice, but the dot matrix printing is readable and consistent throughout the issue. At this writing Commodore is announcing many new products. Exciting times are ahead, and I look forward to participating in them with all of you.

Auto Data Entry

by RHB

There are times that you want to be able to record data in a program for future use. This often occurs when you are doing students grade averages. Your choices are to either store the grades in DATA lines in the program or to write the information in a file. Writing a file requires a cassette tape or disk. The first is unreliable and the second expensive. DATA statements in the program are far simpler but mean that the person entering the grades must know some programming to change DATA lines. You cannot use a simple INPUT since the information entered this way cannot be recorded with the program. A data file on tape solves this problem and is more professional since it does allow the user to INPUT data which is then recorded. These data tapes may get lost and reading these tapes back sometimes is a problem. For large amounts of data a tape or disk is still the best idea. For smaller amounts of data the following technique is useful.

The object of this program is to allow the user to enter grades using INPUT, READ previous grades from DATA statements and figure the average and highest grade. The data which was INPUT is then automatically entered into a line in the program. The user then need only rerecord the program and does not have to worry about changing the program himself.

The program provided may be used in several different ways although grading is the use mentioned. The technique of automatic data entry can be adopted for use in many different situations. The following is a brief commentary on the program with comments provided for all but the most obvious lines.

LINES 220-240

These lines make the program 'smart' so the program knows which ROMs it is working on. Line 220 will set PT (for PET type) to 0 for the old ROMs and 1 for the new. The next two lines set KB (keyboard buffer index - # of keys pressed) and BP (buffer position - where the keys that have been pressed are stored).

LINE 250

This line READs the first piece of data which is always in line 510. It then adds 10 to the number it read. NL then represents the NEXT LINE in which data can be recorded. As the program is used the number in line 510 will increase.

LINES 260-290

Line 270 READs any grades in DATA statements if there are any. If G (the grade) is greater than 100 the program stops READing and goes to INPUT. Line 280 determines the highest grade (H). Line 290 calculates the total of the grades (T) and the total number of grades (N).

LINES 310-340

Line 320 allows new grades to be entered via INPUT and terminates if a grade greater than 100 is entered. Line 330 again checks for the highest grade. Line 340 totals the number of INPUT grades (NG) and sets up an array of input grades (G(NG)). It also totals up the grades and increases the total number by 1.

LINES 350-360,390

The first two lines print out the average grade and the highest grade of all the grades INPUT or READ. Line 390 ends the program if no new grades (NG) were input.

LINES 400-440

Line 400 reprints line 510 on the screen as a DATA line. The DATA is NL on the next line available for grade data next time the program is run. Line 410 prints a DATA line on the screen which has as its line number NL. The next three lines PRINT the grades entered by INPUT separated by commas.

LINE 460

This line PRINTs a DATA line whose number is 10 more than NL. The information in this line is simply 999, the terminator for READING grades.

LINE 470

This line is really the heart of the automatic entering of the PRINTed DATA lines in the program. First, the cursor is homed and left on the first line with a semi-colon. POKE KB,4 fools the PET into thinking 4 keys had been hit. The FOR-NEXT loop then tells the PET that the four keys it thinks were hit were CARRIAGE RETURNS. It does this by placing the ASCII code for carriage return (13) in each of the first four positions in the keyboard buffer. The computer responds by CARRIAGE RETURNing four times and entering the three DATA lines that were PRINTed on the screen.

There are some limits here which might be changed when the need arises. As it is the program will only allow the user to enter as many grades as will fit on one DATA line. The only calculations are the average and highest grade. Also the memory locations referenced must be updated for the new BASIC 4.0. Right now I don't have the necessary memory map to make these adaptations. I hope this explanation stimulates other uses which I hope you will let me know about.

Listing

```
100 REM DATA REVISE
110 REM RALPH BRESSLER
120 :
130 PRINT"(c) DATA REVISE(2dn)"
140 PRINT"  THIS PROGRAM WILL READ GRADES
150 PRINT"FROM DATA STATEMENTS AND THEN
160 PRINT"ALLOW INPUT OF NEW GRADES. THESE
170 PRINT"NEW GRADES WILL AUTOMATICALLY
180 PRINT"BE PUT IN DATA STATEMENTS. THE
190 PRINT"PROGRAM CALCULATES THE AVERAGE
200 PRINT"AND HIGHEST GRADE.
210 GOSUB480
220 PT=PEEK(50003)
230 IF PT=0 THEN KB=525: BP=527
240 IF PT=1 THEN KB=158: BP=623
250 READ NL: NL=NL+10
```

```

260 PRINT"(clr)READING GRADES...
270 READ G: IF G> 100 THEN 300
280 IF G>H THEN H=G
290 T=T+G: N=N+1: GOTO270
300 GOSUB480
310 PRINT"(clr)INPUT NEW GRADES (999 TO STOP)(dn)"
320 INPUT G: IF G>100 OR G<0 THEN350
330 IF G>H THEN H=G
340 NG=NG+1: G(NG)=G: T=T+G(NG): N=N+1: GOTO320
350 PRINT"(clr)AVERAGE ="T/N
360 PRINT"HIGH GRADE ="H
370 GOSUB480
380 PRINT"(clr)(dn)"
390 IF NG=0 THEN END
400 PRINT" 510 DATA"NL
410 PRINT NL"DATA";
420 FOR I=1 TO NG: PRINT G(I);
430 IF I<>NG THEN PRINT "(left).";
440 NEXT
450 PRINT
460 PRINT NL+10"DATA 999"
470 PRINT"(hm)";:POKE KB,4: FOR I=0 TO 3: POKE BP+I,13:NEXT:END
480 PRINT"(home)(23dn)      HIT SPACE TO CONTINUE"
490 GETSP$:IFSP$=""THEN490
500 RETURN
510 DATA 510
520 DATA 999

```

LIPS Contest Winners

This article comes about 6 months late but the students involved should be recognized. The Long Island PET Society held a programming contest and over 30 programs were submitted. They were judged by many different people using a point system. Most of the programs were submitted by elementary school students and prizes were awarded for first and second in graphics and non-graphics categories. All winners were from the Half Hollow school district.

In the graphics category James Moon, a sixth grader at Signal Hill school, took first place for Combat on SRL400. This program requires manual dexterity and careful planning as the user tries to maneuver his cannon to fire at a moving reflector. Second place in this category was taken by Indy 500. This program was written by Robert Bader of Chestnut Hill and David Plass of Paumanok, both are sixth graders. This program simulates the famous car race.

Password Plus took first place in the non-graphics category. Stephanie Boodman and Amy Zelanko, sixth graders at Signal Hill, collaborated on this program. A series of clues is presented to help the players guess a hidden word. In the same category, Joan Gelin, a sixth grader at Chestnut Hill, and Marni Rosenblatt, a fifth grader at Paumanok, took second place. Their program, Hidden Letter has the user try to identify a hidden letter as individual segments are revealed.

Bill Batcher, a computer teacher at Half Hollow, explained that these students learned programming as part of the district's Academically High Aptitude Program. Under this program, gifted students from all of the district's elementary schools spend one day a week at the AHAP center. This program should serve as a model for elementary gifted and talented programs.

Screen Dump

by B. Vince Haluschak

I used this useful routine in a couple of my programs. It could be made more flexible by allowing you to choose what chunks of lines you want printed. Substituting 'FOR SY=A TO B' for 'FOR SY=0 TO 22' and inputting values for A and B should do the trick. The program now prints everything within quotations.

```
10 REM ***** SCREEN DUMP *****
20 REM *
30 REM * DEVELOPED BY SHAWN GLISSON *
40 REM * SAN DIEGO PET USER'S GROUP *
50 REM * ORIG BY D.A. COSTARAKIS *
60 REM * THE PAPER OCT '79 V3 ISSUE 8*
65 REM * CHANGED BY B.V. HALUSCHAK *
70 REM * BOX 490, TERRANCE, BC V8G 4B6 *
80 REM * 604-635-2519 *
90 REM *
95 REM *****
100 :
130 REM ** CAN ALSO PRINT ANYTHING IN QUOTES **
140 :
150 PRINT"HOW MANY LINES PER INCH ? ";
160 GET X: IF X=0 THEN 160
170 REM ** LIMITED TO MAX OF 9 LINES/INCH **
180 PRINT X
190 X=144/X
200 CLOSE 9: OPEN 9,4: CMD9
210 CLOSE 6: OPEN 6,4,6
220 FOR SY=0 TO 22
230 FOR SX=0 TO 39
240 SL=PEEK(32768+SX+40*SY)
250 IF SL<>32 THEN Y=0: GOTO 300
260 IF SL=32 THEN Y=Y+1
270 IF Y=80 THEN SY=22
280 REM ** IF Y=80 BLANKS ON THE SCREEN PRINTING STOPS **
290 REM ** SAVES TIME AND PAPER IF MOST OF SCREEN IS EMPTY **
300 IF SL>127 THEN SL=SL-128
310 IF SL<32 OR (96 OR SL)=SL THEN SL=SL+64: GOTO 330
320 IF (SL OR 95)=95 THEN SL=SL+128
330 POKE 205,0
340 PRINT#9,CHR$(SL);
350 PRINT#6,CHR$(X);
360 NEXT SX
370 PRINT#9: NEXT SY
380 PRINT#9: CLOSE9: CLOSE6
390 END: REM ** OR RETURN **
```

Microview courtesy of Central IL PET Users
by Jim Oldfield

MICROTHELLO, \$10 from AB Computers. Challenging to impossible. A must for anyone interested in Othello. Combination of BASIC and ML makes it FAST. 5 levels of play; level 5 is nearly impossible. Use a blinking cursor to move and watch the PET analyze the board, as it thinks of a move. Many built-in options.

Assembly Language Programming Part II

by Jim Fowler

II. Number System and Loading

The assembly language for the 6502 microprocessor has a vocabulary of 56 'opcodes' or kinds of operations which translate to switch settings in the microprocessor. Many of these can be used in different addressing modes to specify where the next word of data or command is in memory. There are a total of 11 address modes although not all commands use all these modes. This amounts to a total of about 150 useful commands you can give the microprocessor. That is a rich language - far richer than BASIC. When you have written a program you will have to translate it into machine language. The processor 'knows' only binary words composed of exactly 8 bits or binary digits which equal one byte. The pattern of bits in a byte represent a series of electrical signals that are either 0 volts, symbolized by 0, or +5 volts, symbolized by 1. How do we translate our program and then enter the words into PET's memory where the processor can get at them?

Fortunately, this LOADING process is relatively simple. To understand how it works we need to understand the hexadecimal number system (HEX). PET's communication with the outside world could be in any number system. BASIC uses decimal or base 10 but for machine language HEX or base 16 is more efficient, convenient and compact. Besides the digits 0 to 9 HEX uses the letters A, B, C, D, E, and F, to stand for decimal 10, 11, 12, 13, 14 and 15. Thus 10 in hex is 16 in decimal. Frequently we use a dollar sign (\$) to signify a number is in HEX.

Since we are so used to counting and writing in decimal it is hard to think in HEX. To help convert hex to decimal and vice versa a chart is included. In the chart the symbols on the right represent the left digit in a two digit HEX number and the numbers across the top represent the right digit. To convert hex to decimal just find the row and column then read the decimal equivalent. To go the other way find the decimal number in the chart and then look across for the left HEX digit and up for the right one.

What we need now is a way of typing our program into the PET. A large computer system would have as one of its 'utilities' a program called an ASSEMBLER. A good one would allow you to type assembly language mnemonics directly into memory. It would then read what was typed, point out ERRORS then assemble the input you save it and turn it into a machine language program. One advantage to using assembly language is that it is easier to remember what the commands mean since they are abbreviations for their functions. Also most assemblers allow you to give names to commonly used memory locations. The PET actually does this to your BASIC programs but keeps no record of the machine language program it creates. Instead it translates BASIC into machine language piece by piece and once a chunk has been executed it is forgotten.

We can use the Commodore TIM monitor to enter machine language code directly into memory. The new ROM machines have the MONITOR resident in ROM and you call the MONITOR by using SYS1024. It also comes up whenever a machine language program executes a BRK (the BREAK instruction). If you have the old ROM PET then you must load the TIM monitor from tape. Everytime your machine language program 'crashes' you must reload the monitor and your program. I have written a little

machine language program called 'MON' that stays in the second cassette buffer. The listing is included here.

To work effectively in machine language you will find that it is best to add a reset button to your PET. It is hard on the PET's innards to turn the power on and off frequently and, of course, this means you lose your program. NEW CURSOR from International Technical Systems allows you to recover control of your PET without losing your program. In fact, even BASIC programs are preserved. This only works on the new ROM PETs. Another must is a good book on machine language programming. The most comprehensive book is 6502 Assembly Language Programming by Levanthal published by Osborne/McGraw Hill. Also helpful is the PET Machine Language Guide published by ABACUS Software. This book gives many examples and shows how to use PET's built in routines. Finally, hand assembling even small machine language programs is boring and leads to many mistakes. An assembler is very handy and a must for larger programs. There are two assemblers I recommend highly although they are very different. The DASM or TASM package by Baker Enterprises is written in BASIC and has a price tag of \$15. The MAE/TED package by Eastern House Software is written completely in machine language and cost about \$170. Both work. Baker's program is fine for beginner's while very advanced programmers will like the MAE/TED. Both have versions available for tape or disk. See the comments on both of these products in this issue for more details.

Listing

```
100 REM THIS PROGRAM GENERATES 'MON', A LITTLE MONITOR PROGRAM
110 REM THAT STAYS IN TAPE BUFFER 2 FOR OLD ROM PETS. LOAD:
120 REM THIS; PUT CLEAN TAPE IN #1; THEN 'RUN' AN FOLLOW
130 REM PROMPTS. THE TAPE WILL HAVE 'MON' ON IT. LOAD 'MON'
140 REM ONCE AND IT IS THERE UNTIL POWER IS OFF. EACH TIME
150 REM YOU RESET A 'BRK' POKE 540,3: POKE 539,144 WILL SHOW
160 REM YOU ACCUMULATOR, X AND Y REGISTER, STATUS, PROG COUNT
170 REM HI, LO AND STACK POINTER IN THAT ORDER. TO USE 'MON'
180 REM TYPE 'SYS 826'. THEN ENTER ADDRESS AS TWO DIGIT HEX
190 REM NUMBER. 'MON' SHOWS YOU 8 BYTES AND YOU CAN TYPE 'L'
195 REM TO LOAD OR 'S' TO SEE THE NEXT 8 BYTES. RETURN ALONE
200 REM EXITS TO BASIC.
210 FOR I=1 TO 10: READ A,X: POKE A,X: NEXT
220 FOR I=826 TO 1023: READ X: POKE I,X: NEXT: SYS63153: END
230 DATA 2816,77,2817,79,2818,78,249,0,250,11,238,3,247,58,248,3
240 DATA 229,1,230,4
250 DATA 234,234,234,32,213,3,133,7,32,213,3,133,6,32,210,201,160
260 DATA 0,177,6,74,74,74,74,32,189,3,177,6,41,15,32,189,3,230
270 DATA 226,200,192,8,208,233,32,232,3,201,83,208,6,32,177,3,76
280 DATA 71,3,201,76,208,239,160,0,32,210,201,169,145,32,210,255
290 DATA 32,213,3,145,6,230,226,200,192,8,208,244,32,177,3,76,71
300 DATA 3,104,133,12,104,133,11,104,133,10,104,133,13,104,133,14
310 DATA 14,104,133,15,186,134,16,169,10,133,6,169,0,133,7,76,71,3
320 DATA 36,24,165,6,105,8,144,2,230,7,133,6,96,24,105,48,201,58
330 DATA 48,2,105,6,32,210,255,96,32,232,3,201,58,48,3,56,233,7,96
340 DATA 32,202,3,10,10,10,10,133,1,32,202,3,41,15,5,1,96,36,36
350 DATA 152,72,32,228,255,240,251,201,13,208,3,76,139,195,32,210
360 DATA 255,133,4,104,168,165,4,96
```

Hex to Decimal Table

	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
\$0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
\$1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
\$2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
\$3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
\$4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
\$5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
\$6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
\$7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
\$8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
\$9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
\$A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
\$B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
\$C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
\$D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
\$E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
\$F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

SAT/PSAT A Product Review

by Stan Silverman

Type: Software
Model PET: any model
Source: Knell Software
21 Milkbrook Dr
Stony Brook, NY 11790
Price: \$59.95 for 7 programs on 4 tapes

It's a pleasure to write a review of a software package that does what its description implies. Knell Software advertizes and delivers a set of 4 tapes (PET, TRS80, APPLE 2, and APPLE 2 PLUS) which present "a virtually limitless series of questions and answers" based on past SAT examinations "with scoring provided in accordance with the formula used by the College Boards".

In Hempstead, we ordered a set of the tapes and promptly recieved the tapes three weeks prior to the October SATs. Students were informed of the tapes existence, but due to time and staff limitations, no attempt was made to provide teacher reinforcement. The results of the SATs for test group were compared with a control group (normalized as per prior SAT). We noted a 72 point increase for the test group and only a 22 point increase for the control group. The average los time per student was 90 minutes.

The one problem we experienced with this package is the consistency of loadings. The programs contain data files and your heads must be in good order to get reliable loads. The Knell people have indicated they are going to disk and have changed tape suppliers to insure a good load. In addition, they are preparing an educator package which will contain drill exercises, answer analysis, etc.

Knell is a company that is committed to a quality line of USEFUL and enjoyable educational software and I for one will be looking for future releases. They already have an interesting adventure type game.

ML Relocator

by A. Wachtel

Here is a little program I put together when I needed to relocate a machine language program. I know that there are ML programs to do this faster, but nobody ever published 'idiot-proof' directions for this.

```
10000 REM MACHINE LANGUAGE RELOCATOR
10010 REM A. ACHEL, 1980
10020 INPUT"(clr)LOWER, UPPER ADDR OF ORIG PGM";L,U
10030 INPUT"NEW LOWER ADDRESS BOUND";LN
10040 PRINT"(clr)VERIFY: * MEANS ADDR OF A BRANCH"
10050 PRINT"INSTR WITH LENGTH OF 32 2 OR 76 BYTES(cd)"
10060 PRINT"          ORIG          NEW"
10070 PRINT"
10080 PRINT"ADDR CODE      LSB MSB          LSB MSB(cd)"
10090 K=LN-L:FOR I=L TO U
10100 A=PEEK(I):POKE I+K,A:IF A=32 OR A=76 THEN GOSUB 10200
10110 NEXT:END
10200 B=PEEK(I+1):C=PEEK(I+2):X=PEEK(I-1)
10210 IFX=16ORX=48ORX=112ORX=144ORX=80ORX=175ORX=208ORX=240
      THEN PRINT"*";
10220 D=256*(B/256+C)+K:MB=INT(D/256):LB=D-(256*MB)
10230 PRINT I" "A" "B" "C" "LB" "MB
10240 I=I+1:POKE I+K,LB:I=I+1:POKE I+K,MB:RETURN
```

The program is obviously NOT foolproof. What can happen is that there is a byte with a value of 32 or 76 which is not an absolute jump, e.g. a relative branch of such length. When the program encounters this it will misinterpret this as JSR or JMP and modify the next two bytes. Line 10210 will alert the user that this has happened (untried by myself so far!). The reader can decide whether replacing PRINT"*"; by RETURN offers better chances. Remember that then there are eight values which could arise for reasons other than a branch instruction. Does anyone have an idea? The program relocated a 193 byte machine language program in 4.6 seconds and showed 3 JSR and 1 JMP instruction. Without these it would have been much faster.

Speed-Up Explanation

by Warren D. Swan

Here is an explanation for Doug Hennis and all others who wondered why POKE 59458,62 will speed up printing to PET's screen. I hope it is understandable enough! I discovered this trick about six months ago while pouring through the disassembly listing of the PET's BASIC and operating system. I also found that it can cause some PETs to 'hang up' some of the time. To avoid this I suggest you use the two commands: WAIT 59456,32,32: POKE 59458,62.

First, the two locations 59456 and 59458 are locations that reference the 6522 VIA I/O chip in the PET. Location 59456 is an I/O port that, among other things, is used for synchronization of the PET's screen for printing to avoid snow. Location 59458 is called the 'data direction register' for this I/O port. Each bit of the DDR tells whether the corresponding bit of the I/O port will be an input or output bit. Each 1 bit in the DDR means that the corresponding bit of 59456 will be an output bit, while 0 means an input bit. Normally location 59458 (DDR) contains 30, which is 00011110 in binary. Thus

bits 1 through 4 of 59456 are output bits, and bits 0 and 5 through 7 are input bits. Bits of a memory location are always numbered 76543210.

Bit 5 of this port is an input to the computer that tells it when the screen is done being refreshed. When the value of bit 5 is 1, the refresh is done. All PET screen I/O routines check this bit to make sure it is 0, which means 'OK to write on the screen'. Bit 5 of the DDR is initialized at 0 so that bit 5 of the I/O port will be an input bit. However, if we change bit 5 of the DDR to a 1 (which we do with POKE 59458,62), we are changing bit 5 of the I/O port to an output bit. It no longer reflects the status of the screen refresh circuitry, but is always a 0. Thus each time the screen print routine goes to 'wait' for the screen refreshing to be finished, it sees a 0 in this bit position and so it goes ahead and puts the character in screen memory.

The WAIT command merely waits until bit 5 of the I/O port is 0 so that we do not try to change the DDR during screen refresh. This is an attempt to avoid 'hanging up'. However, I am not convinced this is entirely foolproof.

If you want more information, such as why it might cause the PET to hang up at all, please write or call me at: 15933 Grove Ave., Oak Forest, IL. 60452; (312) 687-4876.

(Ed Note: Several sources have told us that using this method is not recommended especially if you intend to sell or trade your programs. At the very least it may cause some PETs to 'hang up'. It can cause two pieces of hardware to 'fight' and may damage some PETs.)

Comments on Volume 3, Issue 4/5

by Francis Turco

I am happy to hear that The PAPER is back in print. You stated in your Editor's Note that "we welcome your comments...". I hope that this does not come across in a negative manner. I have always felt if a magazine yields one or two ideas, it was worth the fee. Your first issue qualifies in that respect. There are, however, a number of things that will bear looking into if you are to continue to improve.

- (1) The PET printer has a poor quality output and there are portions of The PAPER that did not reproduce well.
- (2) The first issue is full of typographical errors. More careful editing is in order.
- (3) The Cross Referenced Memory Map on page 4 is nice. A complete memory map for both old and new ROMs may be found in "PET/CBM Personal Computer Guide" published by Osborne, McGraw-Hill. It contains not only the first 1024 memory locations, but also all the BASIC interpreter locations as well.
- (4) The article on machine language speed on page 10 contains a scrambled line 100. The swapping at the end was typed wrong. In the same article Listing 2 shows a short demonstration program. It seems unnecessary to remove all the spaces. Although the computer does not need the spaces, they certainly would make the listing more readable for human users.
- (5) On page 15, the program listing is very easy to read, but use a form that could be executed.

- (6) On page 17, the article by Gerry Eisner states there are 24 rows from top to bottom on the PET screen. The PET has 25 rows.
- (7) On page 23, line 15, what does the "@" mean in PET BASIC?
- (8) PEEKing at BASIC on page 42 is an informative article and illustrates the use of the USR command nicely. We have already incorporated this idea into our disassembler and have disassembled portions of Microsoft's BASIC interpreter.
- (9) Grouping the ads in the back of The PAPER is a good idea. They are also the easiest part of the magazine to read.
- (A) Page 40 has a survey article on merging/appending programs. The July/August issue of COMPUTE has a machine language routine to append disk files with the Commodore 2040 Dual Disk Drive. No mention was made in your article of techniques for disks.

Response to Fran Turco

Thanks for your comments on our first issue. We need more feedback since its hard to act on the comments of only a few people. Here are some thoughts about your comments.

- (1) You're right about the poor readability and reproduction quality of the PET dot matrix printer. However, noone has offered to donate a letter quality printer. (Any takers?) We are exploring ways to improve the quality of the print.
- (2) Typo's make reading hard and sometimes frustrate the use of an otherwise good program or idea. We try to minimize these kinds of errors. Please understand that only two people type, proofread, retype and paste-up each issue. We hope our recent conversion to WordPro III will help.
- (3) A memory map for all important locations is needed for ALL of Commodore's versions.
- (4) Line 100, page 10 should read:

```
...THEN H=A(I+1):A(I+1)=A(I):A(I)=H:F=1
```

 The spaces in Listing 2 weren't removed. I just type programs that way from habit. We'll try to include spaces for readability.
- (5) To me the listing on page 15 is more readable the way it is. The user may enter the comments as REMs.
- (6) Yes, the PET screen does have 25 lines. Also, there are 256 possible characters numbered 0 to 255. On page 18 in the second paragraph, the loop repeats 40 times. On the same page line 25 should be +I not -I, on the bottom of page 18 line 10 should be E=83 not D=83.
- (7) The @ just sets off a comment. It can be used after GOTO's since the PET will never 'see' it.
- (8) Glad to see that PEEKing at BASIC helped. Remember that it is only needed for old PETs. When Commodore got the new BASIC back from Microsoft they had a problem with PEEK. While correcting the problem they eliminated the check to see if you were looking at BASIC. After correcting the problem they forgot to restore this protection.
- (9) The ads will remain at the back of The PAPER except for some special requests by advertisers.
- (A) Thanks for the comment on appending disk files. With the wider use of disks articles on using them to their fullest will be mandatory.

PET and APPLE II Users

PASCAL

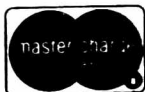
ABACUS Software makes available its version of TINY PASCAL for the users of two of the most popular personal computers.

TINY PASCAL is a subset of the standard PASCAL as defined by Jensen and Wirth. It includes the structured programming features: IF-THEN-ELSE, REPEAT-UNTIL, FOR TO/DOWN TO-DO, WHILE-DO, CASE-OF-ELSE, FUNC and PROC. Now you can learn the language that is slated to become the successor to BASIC.

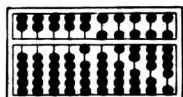
TINY PASCAL is a complete package that allows you to create, compile and execute programs written in the PASCAL language. You can save source and object code on diskette or cassette (PET version only). Comprehensive user's manual included. The manual can be examined for \$10 (refundable with software order).

REQUIREMENTS

PET 16K/32K New ROMS cassette	\$40
PET 16K/32K New ROMS diskette	\$35
Apple II 32K Applesoft ROM w/DOS	\$35
Apple II 48K Applesoft RAM w/DOS	\$35
TINY PASCAL User's Manual	\$10
6502 Interpreter Listing	\$20



FREE postage in U.S. and CANADA
All orders prepaid or COD



ABACUS SOFTWARE

P. O. Box 7211
Grand Rapids, Michigan 49510

PET™ 2001 STUDENT WORKBOOKS

For Classes Beginning to Program

COMPLETE TEACHING UNITS

- Step-By-Step Instruction (Teachers Can Be Beginners, Too!)
- Daily Lesson Plans, Classwork & Homework Worksheets, Quizzes
- Hands-On and Hands-Off Exercises
- Group and Individual Assignments

FEED ME, I'M YOUR PET™ (Beginner-1)

LOOKING GOOD WITH YOUR PET (Intermediate-1)

TEACHER'S PET (Plans, Quizzes, Answer Key)

Workbooks are \$4.35. Quantity discounts available.

TEACHER'S PET is \$4.00 or FREE with workbook orders of 25 or more.

EDUCATIONAL SOFTWARE

- High Quality Teaching Skills and Programming Skills
- Graphics Used to Enhance Learning

REMEDIAL MATH PACKAGE (84 Concepts)

MATH IN SCIENCE SERIES

Signed Numbers, Physics, Primary Primer, Elementary Grade topics and More. We Will Send a Price List.

COW BAY COMPUTING

Box 515 • Manhasset, NY 11030

F.O.B. Manhasset, NY

COMPUTER LAND of Nassau

"The Full Service Computer Store"

Come **SEE** what you've only **READ** about.

Hardware...Apple...PET...Atari...Texas Instruments...Cromemco...Dynabyte...computers for personal and professional needs...Diablo...Centronics...Spinwriter...printers and disk drives to complete a system.

Software...programs for **ALL** the machines we carry...programs for recreation, education and business.

Books and Magazines...Osborne...Blacksburg Group...Hayden...Sybex...Scelbi...or pick up a copy of your favorite magazine.

Watch For New
Store In Suffolk.
Coming Soon!

We Now Carry
Vector Graphics
Products

Computer land
79 Westbury Avenue
Carle Place, NY 11514

CENTERBROOK SOFTWARE DESIGNS

"Learning is Fun"

Remedial Math and Reading

Science, Utilities, and Games

We have over 70 programs in the topics listed above. All programs work in 8K and there are versions for all PETs. Prices for most programs are in the \$10 to \$20 range and special packages are available. New programs are produced every month and custom programming is available.

SPECIALS

(Until February 15, 1981)

Any 6 Math Programs

normally \$90, **now only \$60**

Science Package - biology & chemistry

8 programs sell for \$95, **now only \$50**

NEW Teacher's Grades-normally \$50 - **now \$35**

Basketball Stats-sells for \$50 - **now \$35**

These packages include 5 programs each and completely handle grades and stats. They require a 32K PET, disk and a printer. All programs interact and are meant to be used by new users.

For a catalog and more information write or call:

Centerbrook Software Designs

98 Emily Drive

Centereach, NY 11720

516-585-2402

CBM/PET Computers & Such!

Northern Virginia's single-stop source for Commodore computers, printers, disks, software, and factory-authorized maintenance. Also providing a wide range of accessories and supplies from other manufacturers.

VMS features:

- o PET/CBM by COMMODORE
- o SELECTRIC Interfaces by ESCON
- o Software by MICRO SOFTWARE SYSTEMS
- o Printer interfaces by TYPEWRITERS UNLIMITED
- o Covers & Uncrashers by INTERNATIONAL TECHNICAL SYSTEMS
- o Books, magazines, paper, ribbons, diskettes, cassettes, parts



CONNECT PET/CBM TO PRINTER: Simply plug into the interface and use your PET/CBM computer with popular "standard" printers, or add a video monitor for a second display. All interfaces are assembled, tested, and warranted. No software is required.

TU-65C Use NEC, Centronics, Xymec, or other industry standard parallel printers. Works with disk, other IEEE devices attached. \$129.95

TU-6514 Use RS-232 (serial) printers, like Heathkit H-14. 300 to 4800 baud (please specify rate desired). Includes power pack. \$79.95

TU-PVE Connect your Video Monitor to PET/CBM for second display. Our unique contact extensions leave the user port available for connection of other accessories. \$39.95

ESCON SELECTRIC Typewriters connect to PET/CBM, other computers for quality output. We are Northern Virginia's authorized dealer for this nationally known line. Sales, installation, service! Call for details.

MSS Software that has gained an international reputation for ease of use and quality construction. Reviewed in major national magazines, and available for all versions of PET/CBM computer!

Virginia Micro Systems

14415 Jefferson Davis Highway
Woodbridge, Virginia 22191

(703) 491-6502

VISA/MC
VA add 4%
Dealer
inquiries
invited

Riley Enterprises announces....

TOP Quality Vinyl Cassette Holders
12 Capacity in Black or Brown \$2.80
16 Capacity in Blk, Org or Lime 3.50

Add \$2 for shipping orders under \$20

Other types and colors are special order.

75 Pearson Street
Portsmouth, NH 03801

603-436-6564

Programs!!



- Business Research \$50
- Home & Small Business \$15-40
- Games & Simulations \$15 ea.
- Education (HS) Pack \$15

Send for complete catalog!

HARRY H. BRILEY

- Shopper
- Inventory
- Museum!
- Mansion!
- Pentagon!
- Home Addresser
- Deluxe Addresser
- Dinner's On!
- Fur Trapper
- High Seas

Catalog Info.
P.O. Box 2913
Livermore, CA
94550
(415) 455-9139

AMERICAN PERIPHERALS

First In Education

Printers For PET/CBM

(all cables and interfaces included)

IBM Selectric Model 50 - IBM letter quality output	\$2,200
Xerox Diablo 630 - letter quality, daisy wheel printhead	\$3,200
NEC Spinwriter - letter quality, thimble printhead	\$3,300
Centronics 737 - high density dot matrix, near letter quality	\$1,200
Commodore Mannesmann Tally 8024 - 132 columns, high speed, bi-directional printing, business applications	\$1,750

Upgrade Options

Trade up from your old ROM PET to the new ROMs. You get a large keyboard and an improved BASIC.

Old 8K to New 8k	\$400
Old 8K to New 16K	\$600
Old 8K to New 32K	\$800

Many, Many Other Upgrade Deals Available!

New Items

Interface, software to connect your mark sense reader	\$ 500
Filmstrip: Basic Computer Programming (5 Filmstrips)	\$ 74
Wantanabe MiPlot - flatbed plotter connects to PET	\$1200
TCL PASCAL - disk based version from Commodore, large subset	\$ 300
Tiny PASCAL - from Abacus software, a good start	\$ 40
PET-tran - a simplified subset of FORTRAN for the PET	\$ 100

EDUCATIONAL ACTIVITIES

Bridge the Classroom Literacy Gap. Two series of full color filmstrips with audio cassettes and teacher's manuals present a systematic approach to programming. The programs provide a set of step-by-step instructions to guide students through the BASIC language. These filmstrips can be used at any grade level to teach students the concepts of programming a microcomputer in BASIC. The manuals include program listings, comprehension questions and program assignments. Both sets are designed by Arnold and Seth Friedman.

BASIC for Microcomputers

Includes *Getting Started, Mathematical Operations, Loops and Subroutines, Original Programming Techniques* and *Flowcharting*. Designed for beginners in programming.

5 Filmstrips and Audio Cassettes - \$89.00

Advanced BASIC Techniques

A sequel to the popular series shown above. Topics discussed are *String Functions, Data Handling and Arrays, Computer Graphics* and *Advanced Math Functions*. This series is meant for students who understand the concepts presented in *BASIC for Microcomputers*.

4 Filmstrips and Audio Cassettes - \$74.00

Both series available for preview.

We also have many educational programs for the PET, Apple and TRS-80.

P. O. Box 392
Freeport, NY 11520

WEST RIVER ELECTRONICS

Two New Character Sets

Your PET can now display simple or complex mathematical formulas in standard notation with our *Math Character Set*. With it the computer can display exponents, superscripts, integrals, square roots and much more. Most of the symbols you will need from elementary algebra to advanced calculus are included. $ax^2 + bx + c$

With the *Foreign Language Character Set* the PET has all the extra characters to create text in Spanish, French, German, Italian, Latin, Dutch, Polish, Czech, and Portuguese. (á é í ó ñ < > ¡ ¢)

Both ROMS now only \$60 each postpaid available for all new PETS including the 4000 and 8000 series.

Order from:

In USA:

West River Electronics
Box 605
Stony Brook, NY 11790

(516) 246-4894

In Canada:

Kobotek Systems LTD
RR 1
Wolfville, NS BOP 1X0
Canada

(902) 542-9100

Finally, MAE — A PET DISK-Based MACRO ASSEMBLER/TEXT EDITOR

Works with 32K PET

- Works with 2040 Disk, and can drive 2022/2023 Printer, and/or RS232/20 ma Device thru User Port.
- 100% Disk Based, 100% Machine Language.
- Macros, Conditional Assembly, and a new feature we developed called Interactive Assembly.
- Coexists with Basic, Auto character repeat, Sorted Symbol Table.
- 27 Commands, 26 Pseudo Ops, 5 Conditional Ops, 38 Error Codes.
- Creates relocatable object code on disk.
- Assemble from Memory or Disk.
- String search, search and replace, and inter-line edit.
- Auto line # -ing, move, copy, delete, renumber.
- Labels up to 31 characters — user specifies length.
- Includes extension to PET monitor (disassemble, trace, etc.), Library of PET ROM locations, Relocating Loader, plus more.

Manual, Diskette, U. S. postage — \$169.95
(Requires completion of License Agreement —
Write for details)

Eastern House Software

3239 Linda Drive

Winston-Salem, N. C. 27106

PET/CBM UNCRASHER™

WHAT IS IT? — UNCRASHER™ is a two button device that allows PET/CBM users to regain control of a cursor that's been lost due to programming errors. BASIC programs may be recovered. Machine language programs in the second cassette buffer are not disturbed either.

WHICH PETS/CBMs? — UNCRASHER™ is for all PET/CBM computers that use the "NEW" Version 2 ROMs. (Older model PETS should use the ITS NEW-CURSOR™.)

DOES IT WORK? — You bet!!! See the detailed review of the types of crashes and the concept of recovery in the first issue of *Compute*.

INSTALLATION — Simple, completely illustrated instructions using only a Phillips screwdriver ensure installation in minutes. No soldering or modifications to the computer.

OPERATION — Just follow the simple steps—push the buttons and reset the stack pointer—and PRESTO . . . recovery!

And all this happens without powering the PET/CBM down and up.

WHY UNCRASHER™? — No first class computer such as the PET/CBM should be without this capability. Whether your fancy be programming, business, education, or hobby, hobby, UNCRASHER™ saves you time by uncrashing your slip-ups.

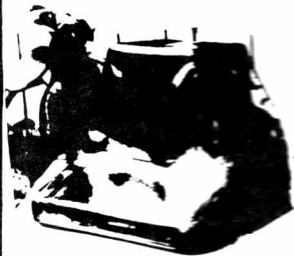
AVAILABILITY — Now in better computer stores, or order direct from ITS, made by the people who brought you NEW-CURSOR™.

INTERNATIONAL TECHNICAL SYSTEMS INC.

P.O. BOX 264 - a WOODBRIDGE, VIRGINIA 22194

CUSTOMER SERVICES -

RICHMOND, VIRGINIA (804) 262-9709



ITS COVERS . . .

A BEAUTIFUL
COMPLEMENT
TO ANY DECOR!

- Attractive, Durable, Leather-Grained Vinyl
- Double Stitching for Extra Strength
- Corded Seams Assure Perfect Fit
- Colors — Gold, Chestnut Brown, Olive Green or Black

PET/CBM — \$18.95
Apple II — \$12.95
Apple Disk — \$8.95

OTHER COMPUTER
COVERS AVAILABLE.
WRITE FOR DETAILS

NEW-CURSOR FOR PET 2001 OLD ROM

INSTANT PUSHBUTTON CURSOR RETRIEVAL

- Reviewed in Issue #1 of COMPUTE
- Restart without touching power
- Machine language programs in second cassette buffer not lost.
- Simple, illustrated instructions
- Installed in minutes — no soldering
- Improved mounting — no sticky tape

\$7.95

UNCRASHER for PET/CBM 2001 - New ROM — \$14.95

INTERNATIONAL TECHNICAL SYSTEMS, INC.

Box 264-a Woodbridge, Virginia 22194

Phone Orders (804) 262-9709



SHIPPING AND HANDLING ADD \$1.00

PET RABBIT



Load, Save, Verify, Execute
8 K in 38 seconds versus
PETS 2 Minute 45 seconds,
plus more!

High-speed Cassette Routines work with 8K, 16K, or 32K new ROM PETS which have the new Commodore cassette deck (like the external version which sells for \$95.00). Note: If you have a new ROM PET with the old style lift-top deck, everything but the high-speed cassette routines will work.

- Auto repeat of any key held down, toggle character set.
- RAM Memory Test, convert #'s to hex and decimal.

12 Rabbit Commands

Note: Rabbit is 2K of machine code at \$1800 for 8K PETS, \$3000 or \$3800 for 16K PETS, or \$7000 or \$7800 for 32K PETS. (Specify one of the 5 versions.)

Cassette and Manual — \$29.95 (Add \$5.00 for foreign)

Eastern House Software

3239 Linda Dr.

Winston-Salem, N. C. 27106

DISK MASTER

\$12.00 On Disk

The first disk cataloging program to be introduced for the Commodore PET/CBM with a 2040 disk. Now you can easily catalog as many as 140 diskettes, forming a large master directory on a single diskette. The program automatically reads the directory blocks of any disk being cataloged, so no typing is involved. File searching, disk ID lists, and other useful functions are also provided.

RIBBONS

for your PET/CBM 2022/2023 printers

\$20.00/Dozen

VISA/MC Accepted - NJ Residents Add 5% Tax
SASE For Current Program List

PROGRAM FINDER

\$7.50/Tape or \$9.50/Disk

Provides easy means of locating particular programs or groups of programs. Each program is cataloged by name, type (business, games, utility, etc.), author, features, (joysticks, sound, printer, etc.), etc. Two versions available using tape or disk data files depending on your system.

BAKER ENTERPRISES

15 Windsor Drive
Atco, NJ 08004

Programming Contest

The Long Island PET Society and Herkimer County BOCES are sponsoring a computer programming contest open to all secondary school students in New York State. The objective of the contest is to increase interest in programming at all levels of competition. Each competing secondary school will choose its best two teams of two students. These students will then compete on a regional level with winners competing on a state level. Contacts have been made in the following areas: Nassau, Suffolk, Westchester, Herkimer, North Country, Buffalo, Rockland-Sullivan-Orange, Southern Tier, Syracuse, Rochester, and Capital District. Details will be announced through the state science and math teacher's publications, the NYSREDS journal and The PAPER. Further information can be obtained by writing LIPS, 98 Emily Dr., Centereach, NY 11720.

The contest on all levels will consist of 5 five minute problems of increasing difficulty. Winners will be determined by the lowest cumulative time to solve these problems.

The PAPER

Box 524

East Setauket, NY 11733

THIRD CLASS POSTAGE
PAID
PERMIT NO. 96
EAST SETAUKET, NY